Investigation On The Algorithm Efficiency Of Generating The Interlocking Control Table Using The Compositional Approach

Mahtab Ibnazia

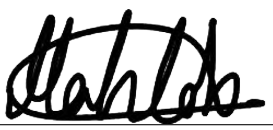
A dissertation submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Master of Science in Engineering.

Johannesburg, February 2022

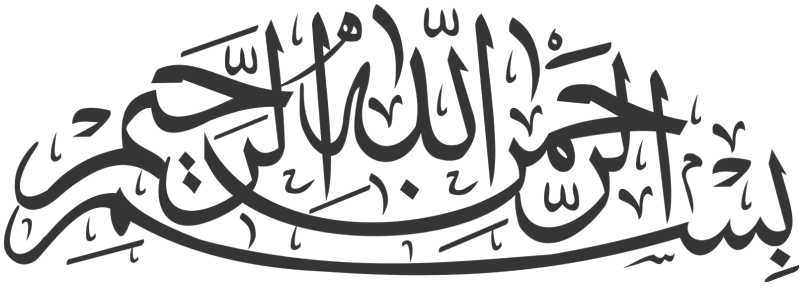
Declaration

I declare that this dissertation is my own, unaided work, except where otherwise acknowledged. It is being submitted for the degree of Master of Science in Engineering to the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination to any other university.

Signed this Mon day of 28 Februray 2022



Mahtab Ibnazia



In the Name of Allah, the Most Gracious and the Most Merciful.

Abstract

The control table is the most essential document that specifies all the safe movements of the trains on the railway track and prevents any possible conflicting movement of trains on the railway track. The work presented extends and contributes to research into train authorisation systems which are also known as railway signalling systems. It focuses on the automatic generation of the control table using a compositional approach. Although previous work in this area has produced various conventional approaches to generate and validate the control table automatically, there has not been sufficient investigation into generation of control tables using the compositional approach. Conventionally the control table is generated by linking the railway signalling elements together according to their geographical position in the station layout and developing algorithms that produce the conditions of the control table. In the compositional approach the large station is divided in sub-sections and algorithms are developed to produce the conditions of the control table using these sub-sections.

In the research presented, the complexity of the algorithm on generating the control table applying the compositional approach is analysed. Software algorithms are developed for both the approaches, namely the conventional approach and the compositional approach. These algorithms are applied on a number of station layouts by gradually increasing the size of the station layouts. It is transpired through the experimental results that generating the control tables using compositional approach is approximately 48% more efficient than the conventional approach for the station layouts that is used in this research. Hence it can be envisaged that the control table generating program using the compositional approach will become more efficient as the larger and more complex station layouts are used. This research work contributes a unique and valuable insight into the control table generation in railway signalling that also combines the knowledge of software development which is in accordance with demand of the modern signalling technologies.

This dissertation is dedicated to...

To my lovely family...

Specially to my mother who has always encouraged me to achieve the greater heights and also to my wife who has supported me throughout my difficult times.

Acknowledgements

I would like to thank Professor Ken J. Nixon for his enthusiasm and the insight which he brought to this work.

I would also like to thank Professor Barry Dwolatzky for his support and guidance.

Special thanks are extended to Professor Ahmad Mirabadi for his advises on the various field of research on railway signalling.

Contents

List of Figures

List of Tables

Nomenclature

ASM	Abstract State Machine
BMC	Bounded Model Checking
CENELEC	European Committee for Electrotechnical Standardisation
CHM	Communicating Hierarchical Machine
CPU	Central Processing Unit
CSV	Comma-Separated Values
CTC	Centralised Traffic Control
DSL	Domain Specific Language
EURIS	European Railway Interlocking Specification
FSM	Finite State Machine
GDL	Geographic Data Language
GPL	General Purpose Language
GRACE	Graphical Requirement Analysis CENELEC Engineering
HOL	Higher-Order Logic
IDE	Integrated Development Environment
LARIS	Language for Railway Interlocking System
MB	Megabyte
ms	Millisecond
OOP	Object Oriented Programming

RSM Recursive State Machine

SIL4 Safety Integrity Level 4

SMT Satisfiability Modulo Theories

SSI Solid State Interlocking

TCO Traffic Control Officer

UML Unified Modelling Language

Glossary

Absolute Block Section	A signalling system of controlling train movements whereby only one train is allowed into a block section at a time.
Fail-Safe	A design philosophy that incorporates any expected failure of the system in design phase and ensures in the event of the failure the system operates in its most restrictive state
Flank Protection	Providing protection from overrunning train movements approaching on a point set where tracks are converging
Interlocking	A signalling system that controls the signalling equipment, for instance, signals and points, to safely authorise the trains and prevent any unsafe train movements
Overlap	An extra distance of clearance provided beyond the danger signal in case the driver does not apply brake on time
Point	A device that allows the train to move from one line to another
Route	The path along a section of railway track from the departure signal to the destination signal
Section	This refers to the portion of railway between two signals
Signal	A device that displays visual instructions to the train driver

Signal Aspect	Any valid visual indication of a signal by displaying a single colour light or a combinations of colour lights as displayed to the train driver
Track	This refers to the section of railway track which is equipped with the detection device that enables to detect the track vacancy status

Chapter 1

Introduction

Many countries suffer great economic losses due to poor infrastructure of the transportation industry despite having natural resources in abundance. Railway is the most effective mode of transport due to the low friction between the rail and wheels of the train [1]. It is such an important mode of transport for state economy mainly due to two factors, these are, a single train is able to commute a large number of passenger and a high volumes of goods and commodities over a far distance with much ease compared to other land transports. Therefore railway authorities continually aspire to upgrade railways for the economic growth of the country. Subsequently the improvement of the train authorisation system is inevitable for an advanced railway infrastructure. Without an adequate train authorisation system the railway infrastructure cannot be used to its maximum capability. Train authorisation system is also referred as a signalling system.

Signalling is one of the most critical and complex engineering discipline in the railway industry. The main objective of signalling is to move the trains from one place to another safely and effectively. The highest safety level critical system needs to be implemented for the train authorisation system due to the strict requirements of the railway signalling system [2]. The signalling system also ensures the most effective use of the railways by permitting the most appropriate train speed, train length and train mass on the network without compromising the safety requirements [3].

The railway interlocking resides in the core of signalling. Interlocking is the equipment that is responsible for the safe train movements and protection from any possible conflicting movements of trains. Railway organisations have been using various types of interlocking through the passage of time. Initially mechanical interlocking, relay-based interlocking, and hybrid interlocking were the most commonly used interlocking

systems. Electronic interlocking systems are slowly replacing all previously existing interlocking systems [4]. Interlocking table state all possible legal routes of the trains over the railway network. Interlocking table is also referred to as a control table or condition table.

The control table is a formal document which is presented in table format to record all necessary conditions for the safe legal train movements. Traditionally the control tables were recorded and checked manually by the railway organisations. The idea of generating control table automatically originated because writing the control table manually was a time consuming task. Moreover, individuals were more prone to make mistakes while recording the contents of the control table. Research groups throughout the world applied various technique to automatically generate and verify the control table [5].

The purpose of this research is to investigate on finding the efficient approach to generate the control table automatically. The research was conducted on two approaches to generate the control table, namely, conventional approach and compositional approach. The dissertation is prepared from the investigation of the research. This dissertation proposes to auto-generate the control table using the latest compositional approach. The structure of the dissertation is as follows:

Chapter 2 provides an overview of the work addressed in this dissertation. The research question is described in detail. The approaches to investigate on finding the best possible answer of the research question are provided. A brief outline of the research analysis that involves the comparison between two approaches is mentioned in this chapter.

Chapter 3 presents the literature survey on the research area. Previous research on the control table generator and verifier is provided which demonstrates the reason for automating the process of the control table generation. The outcome of the previous research work is presented. The focus of this research is clearly outlined in this chapter.

Chapter 4 discusses background information associated with railway signalling discipline. A brief description is given on the history of the railway signalling. The chapter explains the purpose of the signalling in the railway environment. A complete elucidation of the major signalling equipment and functionality of these equipment are provided. Essential signalling rules and regulations are outlined in order to explain

the safety critical train authorisation system. Finally, the fundamental principles of the control table for permissible train movement is illustrated in detail.

Chapter 5 discusses the proposed methodologies for automatically generating the control table. These methodologies includes conventional approach and compositional approach for interlocking control table generation using a software program. This chapter clearly elucidates the processes of software development. It further elaborates the crux of the various software techniques that was used to develop the program.

Chapter 6 provides an analysis of the algorithms of auto-generating control table programs. The analysis was performed in data structure point of view. The kernel activities of the algorithms are explicated by making use of activity diagrams. The computational efficiency of the algorithms were investigated that assisted in gaining a broad understanding about the pros and cons of both algorithms, namely conventional and compositional approaches. The theoretical analysis enabled the development of general formulae which describe the performance of algorithms that assisted in predicting the experimental results. Finally the experimental results were documented in this chapter.

Chapter 7 discusses the findings of this dissertation and provides a conclusion to the work. This chapter also provides the recommendations for future improvements.

Appendix A illustrates all the signalling station layout whose control tables were generated automatically. There were a total of seven layouts from which four of those layouts were standard station layout that aided to identify the standard compositional sub-sections where as the fifth to seventh layout were large and complex station layout that contained the standard sub-sections. This appendix also presents the output text files of the auto-generated control table. All output files that were generated using the programs of conventional approach and compositional approach are documented.

Appendix B presents the Unified Modelling Language (UML) class diagrams and the relationship diagrams between the classes for the control table generator programs. UML class diagram outlines the classes and provides more insight regarding the structure of programs. Relationship diagram illustrates all interactions between the classes.

Appendix C provides the performance of the control table generator programs in terms of the various computational resource usage, for instance, Central Processing Unit (CPU) execution time and memory usage. The Performance Profiler tool in

Visual Studio was used to investigate various performance categories to evaluate the efficiency of the program.

Appendix D records the execution time of the entire program and the time taken by the algorithm to generate the control tables. A batch file was created to conduct the multiple runs of the executable files of the control table generating program on the various station layouts.

Chapter 2

The Strategic Approach

This chapter provides an overview of the work addressed in this dissertation. The research question is described in detail. The approaches to investigate on finding the best possible answer of the research question are provided. A brief outline of the research analysis that involves the comparison between two approaches is mentioned in this chapter.

2.1 Research Question

The objective of the control table is to list all the possible safe and legal train routes of the station in a format of a formal document. Mostly the recording of the control table is carried out manually in a tabular format. Many research groups are investigating and searching for new ways to populate control table automatically [5]. Various formats of control table used throughout the world by the different railway organisations. These is no standardised format of the control table. The format and content of the control table varies among the railway organisations. Although there is no standard format of control table, the principle of train working rules remain the same [6]. Moreover, general safety related rules for the train authorisation must be maintained. Subsequently all the safety critical contents of the control table must be tabulated within the chosen format of railway organisation.

In a more conventional approach, all the signalling elements of the station are connected with each other within the program. The elements are connected according to its geographical position in the station layout. The algorithm is developed whereby the program traverses from one element to another searching for the correct content

of the control table and placing them in the correct places. Signalling researchers found that the computational resource usage, time and memory space, are fairly large for verifying the complex and large stations when using the conventional approach [7]. This directed the researcher to investigate on exploring new approaches to verify the control table.

It was observed through the previous literature that the compositional approach has not been used for generating the control table and it has only been used for verifying the control table. The main focus of the previous researches was to evaluate the performance of the program rather than evaluating the efficiency of the algorithm. The question to be answered in this research is: "Does the use of the compositional approach to generate the interlocking control table for large station layout lead to the significant increase in the algorithm efficiency?"

2.2 Research Approach

A software program will be developed to generate the automatic control table. The research will be performed to generate control table using two approaches:

- Conventional Approach
- Compositional Approach

Previously various research groups invested plenty of time and resources to generate the control table automatically [1, 2, 8, 9, 10, 11]. A number of different approaches were presented to generate and validate the control table [5, 6, 7, 12, 13]. One of the most conventional approach has been generating the control table by linking the signalling elements to each other [6]. In this research the compositional approach of generating the control table will be investigated. Compositional approach is fairly the latest approach in the signalling environment which is used to validate the control table rather than generating the control table.

Two separate algorithms were developed to generate the control table using two different approaches. Namely, the first algorithm utilised the conventional approach and second algorithm utilised the compositional approach. An object-oriented programming style was applied to develop these software program. Some of the most

state-of-the-art software and programming techniques were used to ensure an effective program was developed [14, 15, 16, 17, 18].

2.3 Research Analysis

The dissertation provides a thorough analysis of the research work. Initially a theoretical analysis applied to predict the expected results of the programs. Thereafter actual experimental analysis were performed using the actual results from the program.

Since the propose of the research is to measure the efficiency of the algorithms, hence a algorithm efficiency measuring technique needs to be selected. In data structure the algorithm efficiency is measured in terms of complexity of the algorithm. *Big-O Notation* is a great measuring technique to analyse the complexity of the algorithms. This research used the *Big-O Notation* to evaluate the efficiency of the algorithms of conventional approach and compositional approach.

Both programs with different implementation, conventional and compositional, were applied on several signalling layouts. Commencing with simpler and easier layout to gradually increasing the complexity of the layout, therefore varying the layout inputs the programs were tested to generate control table automatically. Through this experiment the actual Big-O calculations were recorded for the programs with both methods. This enabled a means to compare the results. It was observed that the algorithm developed using the compositional approach is more efficient than the conventional approach.

Chapter 3

Literature Review

This chapter presents the literature survey on the research area. Previous research on the control table generator and verifier is provided which demonstrates the reason for automating the process of the control table generation. The outcome of the previous research work is presented. The focus of this research is clearly outlined in this chapter.

3.1 The Inclination Towards Automation

The railway transport industry requires to establish the safest system for the train movement since any malfunctioning of the system may cause serious consequences for instance, loss of human life, damage of goods and commodities, etc. The interlocking resides in the heart of the safety critical railway system and guarantees that trains are authorised in the safest manner. The control table lists all the key requirements of the interlocking system. It specifies all permissible routes for a station on which trains are allowed to traverse safely.

The most important step is to draft the control table of the station for designing the interlocking system. The control table of a relatively smaller station can be drafted and checked easily. However, when the station layout becomes large and complex then the drafting and checking of the corresponding control table becomes more complicated. It is also certainly a most cumbersome task to completely draft and check the control table manually hence easy to make mistakes. Therefore the railway authorities from all over the world is focusing on developing the control table automatically.

The purpose of the auto generation and verification of railway control table is to generate and verify the control table of a complex interlocking system effortlessly. The automatic generation and verification of railway interlocking control table is an active research topic which is investigated by several railway research groups throughout the world [14, 19, 8, 5, 6, 7, 12, 10].

3.1.1 Interlocking Implementation Methods

Primarily the relay interlocking circuits were implemented as the Boolean equations in electronic interlocking. There is a basically one to one relationship between the relay circuits and the software Boolean variables. It makes it easier for a signalling engineer to validate the software Boolean equations since these are merely representations of traditional relay circuits. This method gives the signalling engineers sufficient confidence concerning the safe working of the system. There are two main methods applied to implementing the model of interlocking systems. They are:

- Functional modelling – Centralised point of view
- Geographic modelling – Distributed point of view

The functional modelling and geographical modelling are explained in details below:

1. **A functional modelling**

   In functional modelling the interlocking systems use a central database where all the rules of the interlocking logic are stored [10]. The design methodology applied in this model is based on functions, such as the route setting functions, the point's position switching functions [10]. These functions are designed based on a control table, which ensures all of the conditions are satisfied before a signal can display the proceed aspect to the train diver [10].

   The central database stores the control table logics of a station layout. In this model the actual implementation has limitation:

   - The control table is not generic and it changes depending on the station layout;
   - Normally these Boolean equations are in huge numbers and very difficult to verify.

The actual position of the signalling equipment in the station layout is ignored in this design methodology [11]. There is no direct correlation between the geographical position of the signalling equipment and the functions of the central database that controls the interlocking [11]. Therefore it is a difficult task to identify the parts of the system simulated by the external events [11].

2. **A geographical modelling**

   The geographical modelling based on the direct positioning of the signalling equipment in the station layout. In this approach the interlocking rules are distributed to objects modelling the geographical placement of physical elements [10]. The geographical structure enables a complete interlocking divide into independent and distributed units of interlocking that can be individually implemented and physically placed close to the relevant equipment. EURIS language (European Railway Interlocking Specification) is a visual and graphical specification language for railway control systems which is designed as per the geographical specification [20]. This language makes it possible to design a component based interlocking system.

   One of the most critical benefits of geographical modelling is revalidation. The station layout can be changed for various reasons, such as maintenance work, or for extensions of the station yard [10]. These changes may occur several times throughout the lifetime of an interlocking system which may span several decades [10]. Computer based interlocking systems have the obvious advantage over relay-based ones in the sense that any changes can be tackled by simply changing the software [10].

   The verification & validation activities of a safety critical software is a costly exercise due to the strict guidelines to be followed in the development process. Therefore revalidation of software for any changes of the system is not a feasible solution considering the financial losses suffered due to the modifications [10]. Alternatively the better solution is that changing a part of the layout affects only those software objects that are geographically related to the changed ones and only the affected ones need to be validated [10]. This approach has broadened the field of design of interlocking including latest the object oriented approach [10].

3.1.2 Software Development Techniques and Methods for Electronic Interlocking

Interlocking is the heart of the Signalling system and it performs a central role granting the safety of the overall system. Standards of developing software programs for safety critical systems are very firm and not flexible for any changes. For this reason, the international governing bodies prescribed standards with high safety levels for railway safety-critical systems. There is no room for taking shortcuts for the development of software that utilised in a safety-critical environment [21].

Agile methods ensure that development of the software happens systematically to improve the quality of software programs in hazardous and safety-critical environments [21]. Agile life cycle models are considered for software development in railway environments which follows an iterative process of development. Each iteration of the complete development cycle comprises critical steps such as requirements, specifications, integration and testing [21]. Agile methods provide an opportunity for the software to be validated at an early stage while considering the feedback and ensuring the necessary changes into the following iterations. Agile processes are suitable for software development in a safety-critical environment [21].

The waterfall life cycle model applies a planning-driven process model for software development. It extensively emphasises on requirements, planning ahead of project implementation [22]. This model utilises a phased approach in developing software. These phases are clearly defined and must be strictly followed throughout the development process [22]. The verification and validation phase ensures output results are compared to the required expected results. The waterfall processes checks and monitors the quality of software produced during each phase [22].

Software was developed using specific domain language in the earlier era of computerised electronic interlocking [1]. The latest trend is to utilise the model code translated to multipurpose languages as ADA, C or Lustre [1]. It will be impossible to describe all the engineering language that has been used for the production of electronic interlocking. A short description is given below for the most popular engineering language used for electronic interlocking:

1. **Geographic Data Language (GDL)**

   The solid state interlocking (SSI) systems make use of GDL. The site-specific

behaviour of the interlocking is detailed using this language [1]. The generic SSI program depends on configuration data. This data specifies the valid conditions of route setting for the specific track layout controlled by the interlocking [1]. SSI interlocking module stores a database with Geographical data that contains the necessary geographic information of the track layout [1].

2. **CHILL**

   CHILL is designed for implementing large and complex embedded systems which are strongly typed and block structured language [1]. CHILL was designed to perform following functions [1]:

   - Extensive compile time checking to improve reliability and efficiency of runtime;
   - Encompass the required range of applications by being flexible and powerful;
   - Exploit a variety of hardware;
   - Facilitate piecewise and modular development of large systems;
   - Provide built-in concurrency and time supervision primitives for real-time applications;
   - Generate highly efficient object code;
   - User friendly and quick to learn.

3. **European Railway Interlocking Specification (EURIS)**

   A graphical method of designing interlocking systems. EURIS comprises four sub-languages. Three out of these four languages are used to model the topology and the routes while the fourth language models the functions of actual entries, for example, the operation of a signal, etc. [1]. The EURIS program was translated into Petri net, which has a precise mathematical meaning, to verify the safety properties [1]. Language for Railway Interlocking Systems (LARIS), a second language, was defined due the lack of formal definition of EURIS [1].

4. **Graphical Requirement Analysis CENELEC Engineering (GRACE)**

   GRACE, the graphical and object-oriented mode of EURIS, perfectly compatible with the geographical approach of SIMIS-W interlocking that allows any signalling engineer to become instantly familiar with the tool due the similarity of the model with relay design [1]. GRACE also facilitates a simulation package for verification of the implemented interlocking structures [1].

5. **Ladder Diagram**

   Many electronic interlocking systems make use of ladder logic, a graphical representation of Boolean equations, particularly customised for reactive systems [1]. This low-level language of Boolean equations is easily verifiable without much effort [1].

3.1.3 Specification and Validation of Electronic Interlocking

The specification and validation of systems is a primary and mandatory task for the approval of railway signalling systems. This implies that the software validation for such systems must be undertaken with the utmost importance. But it is impractical to repeatedly carry out the exhaustive testing of the railway interlocking system software. These systems are to be validated for all sorts of errors that may cause the system malfunction. However, some errors are very difficult to simulate in a testing environment and as a result faults and malfunctions often go undetected. Hence defects may still remain in the system that are tested and verified in a conventional manner and cause accidents in operation.

A number of formal methods used to establish system specifications and verify safety properties of railway interlocking systems. A number of formal languages, such as Petri net, CSS, CTL, CSP, B, VDM, SPIN and Statechart, were used to generate and verify the control tables.

3.1.4 The use of Domain Specific Languages in Interlocking

Domain specific languages (DSLs) are exclusively developed for the use of the particular domain as opposed to the general purpose languages (GPLs) [23]. GPLs are not specific to any domain and they do not have any special features for any particular domain [23]. DSLs are extensively used to develop the railway interlocking systems [24, 23].

L.H. Vu clearly pointed out the differences between DSLs and GPLs. The main differences between DSLs and GPLs is that DSLs specialise in expressiveness since they need to express the artefacts of the domain concepts and notations in ease while GPLs strive for generality [23]. However these artefacts can also be expressed using GPLs. Some of the advantages of the DSLs are [23]:

- DSL descriptions are easily readable, hence easy to understand.
- DSL prevents certain errors from being made, hence they are prone to mistakes.
- It is straightforward to use the domain knowledge in DSLs , hence it boosts the productibility.
- DSLs are developed in such a way that the concepts and the notations are reusable.
- DSL offers efficient analysis and makes the verification process easier.

Some of the drawbacks of the DSLs are [23]:

- Intensive efforts required for the development of DSL while it can express limited problems.
- DSL need to extended if the application need any extra features.
- The development of DSL requires both the domain knowledge and software development knowledge.

Nevertheless, nowadays GPLs such as Ruby, Scala and F# support the development of DSL tools [23].

3.2 The Latest Researches on Automation

The main objective of an interlocking system is to safely move trains within the railway network which under its own control. As a result railway authorities shall apply the safest interlocking system. According to the CENELEC standard the highest Safety Integrity Level 4 (SIL4) is required to operate such vital safety critical railway system [25].

The program that develops interlocking system comprises of the safety critical software. The safety case of the software guarantees the safety integrity of the interlocking system which controls all signalling elements of the railway such as signals, points and train detection equipment [9]. The interlocking must be able to clearly determine safe routes within the network and avoid conflicting routes. The initial step of designing

the interlocking system is to determine all possible safe routes of the station [9]. The purpose of the control table is to correctly record all the valid routes of the station.

If mistakes are made in construction of the control table these mistakes will pass on in the design of the interlocking system that may result in fatal accidents [9]. However, if mistakes were found in more advanced step of designing the interlocking systems then the control table needs to be reconstructed and interlocking system needs to be redesigned. This will require double the effort for same task. This is for the best interest of the designing team to get the control table correct the first time around.

The development of the software of interlocking systems must meet the requirement of highest SIL4 standards to be able to certify as a safe interlocking system [7]. The CENELEC EN50128 standard highly recommends to make use of formal methods and formal verification for the development of the SIL4 interlocking software [19].

Model checking techniques are used for verification of control by the formal method community. This technique fails to produce satisfying results on large interlocking systems due to the state space explosion problem [14]. Conventionally the interlocking software generation and verification process consists of mainly two modules [19]:

1. The first module refers to the generic part of the software which is same for all the stations
2. The second module refer to the configurable part of the software which depends on the station layout and therefore it is specific for this layout.

The generic software is developed and verified for once and not to be changed afterwards. If any changes need to be made of the generic software then the entire verification process need to be followed for safety of the system. Once the generic software is developed and verified then it is repeatedly used for all other station layout.

The process of an interlocking entails firstly specifying the station layout that is to be controlled and thereafter the control table of the station layout to be specified. The control table describes the permitted legal train routes in the railway network and the control rules that the interlocking must adhere to. The configurable software and generic software is integrated to develop the interlocking. A verification tool is developed to verify that the control table is error-free, while other verifying tool is developed to verify the safety of the system [19].

For this reason the signalling research groups has realised the importance of verification of control table. This led to the development of automated software techniques for verification process. Some recent attempts by several research groups has shown a trend to apply various model checking verification process. The model checking refers to the verification process that is used to verify a given interlocking model of a station that fulfil the expected behaviour according to the specification [26]. Model checking is performed by making use of state space equations extensively [26].

The control tables are verified by converting the control table conditions into interlocking design models and the safety of these models formally verified by model checking. Formal verification of interlocking system using model checking techniques require unreasonable computational resources and causes the state explosion problem [26]. Research has shown the exponential growth of the state space as the number of elements increase in the station [19, 26].

The latest research in model checking has brought new techniques to verify the model of the interlocking systems controlling the larger and complex station layouts [7]. Others have applied abstraction techniques at the domain modelling level before applying the model checking for large stations [7]. Some used very efficient techniques such as Bounded Model Checking (BMC) and k-induction [7]. A method found to be successful to verify interlocking systems of large station layout in the use of a combination of inductive reasoning and Satisfiability Modulo Theories (SMT) based Bounded Model Checking (BMC) [14]. This method of model checking performed much better to verify a multi-station interlocking controlling the entire line in comparison to the other tools which failed to conclude the verification within the given time and resource available [14]. Research has shown the verification using these methods reached the limits of time and memory showing that larger systems cannot be addressed anymore [14]. These verification methods need to therefore be further improved.

However, there are still possibilities of having data errors in the control tables that cannot be found by model checking [19]. Research groups has introduced static checker which is able to find all kind of data errors in control tables which does not lead to safety violations and therefore cannot be found by the model checking. The static checker takes the station layout and the control table as inputs and verifies the accuracy of the control table [19]. If the static checker finds any error then it suggests how it can be fixed [19]. Static checker is relatively faster and uses much less memory than model checking to find the other errors [19]. Static checking cannot be used separately, it should be used in conjunction with model checking [19].

Numerous modelling techniques were used to design and develop the interlocking tables. Some of the most illustrious techniques are thoroughly explained in sections 3.2.1 to 3.2.4.

3.2.1 Computer Based Algebra

E. Roanes-Lozano et al. presented an expert system for designing the routes of a railway network that is independent of the station topology [15]. The system uses the topology of the network as the input data and determines all possible routes of the provided section. Thereafter it also verifies the safety of the proposed new routes. This system utilises a powerful list handling facility provided by the computer algebra system, Maple [15].

Since the track arrangement is entered into the system as input data, hence it is capable of deriving all routes that begin in a given section and all routes that are an extension of a given route [15]. The system arranges the route as per their lengths and estimates the travel time of each route [15]. The main objective of this system is to display the option to the station master, who has to make the final decision on which route to select. The station master's choice will depend on various factors, for example, the total travel time of the train, the interim and final destination of the train, and etc. [15].

The system is able to make the station master's job much easier by optimising the route as per his/her requirement [15]. One of the shortcomings of the system is to determine the detailed position points of the route and to impose the aspect of the signals [27, 15]. The second improvement is advisable for the program is to translate this model into a classic language like C for the sake of speed and portability [27, 15].

3.2.2 Ladder Logic

K. Kanso et al. have written a software which verifies the railway interlocking system using SAT solver technology [12]. They have applied this software to the interlocking system of a small UK railway station [12]. Software is developed using ladder logic [12]. Ladder logic is a graphical representation that defines sets of Boolean equations [16]. The interlocking systems are implemented using these Boolean equations. Essentially the electrical circuits of the relay-based interlocking are converted to Ladder logic

diagrams. The main properties of the ladder logic are variables and the Boolean assignments of these variables.

The generation of the ladder logic diagram from its control table is a complex and meticulous task for even a small railway station. Therefore it required extensive human labour to derive the ladder logic diagram of even a small station layout [16]. Hence experienced signalling engineers are needed to look for errors from the ladder logic diagrams as per the signalling principles [12].

K. Kenso et al. have developed a verification software that incorporates signalling principles in a ladder logic program [12]. They have developed a program that inputs the signalling principles in a formal language, and produces safety conditions for which the ladder logic are verified [12].

The software was applied on a relatively smaller station with two platforms and one railway line with two tracks feeding into it [12]. This model required 331 assignments and 599 variables in ladder logic programs where the running time of the SAT solver never took longer than a couple of seconds [12]. The research group concluded that the verification of railway interlocking for at least a smaller station is feasible [12]. However they predicted/ envisaged the computational complexity will grow exponentially for the interlocking with a large number of assignments [12].

3.2.3 State Machine

Railway interlocking requires a safety critical system to reduce the risks to minimum acceptable level by applying regulations and standards [17]. CENELEC 50128 strongly recommends finite state machines (FSM) for modelling and verifying the interlocking control algorithms [17]. A railway system goes through different phases during its development life-cycle where verification and validation (V&V) is the most critical phase [28]. Hence it is one of the most time-consuming tasks for developing the system [28].

The scientific community has developed and applied numerous formal methods and techniques to the railway control applications in past decades [28, 29]. Although these methods are not popular industrially, FSMs are widely used to model the control systems. Statecharts are extensions of FSMs which offers more features, such as, hierarchy, concurrency and communication among concurrent components [30].

Statecharts that are able to integrate with UML 2.0 which is more popular compared to the other variants of statecharts [31]. UML state machines perform parallel execution using composite states and regions [28]. Communicating Hierarchical Machines (CHMs) are a variant of statecharts that introduced the idea of collection of FSMs modules having nodes and boxes [32]. Each box is associated with one or more instances of the modules and transition entering a box corresponds to a call to the associated modules [28].

The introduction of the notion of module enabled the Recursive State Machines (RSMs), where a module can recursively call itself [32]. STATEMATE and Stateflow are the variant of statecharts that based on commercial specification environments [33].

B.T. Celebi et al. modelled and verified control tables using abstract state machines (ASM) [17]. A generalized software was developed to be able to automatically check the model through NuSMV software [17]. A general structure with common properties of the station layout and the interlocking table were analysed which aided to obtain the ASM model structure. The ASM model of the interlocking table was systematically converted so that it could be used with NuSMV software. This established a system which allowed the checking of the interlocking automatically [17].

Clark has given a formula that describes the model checking where M is Kripke structure and f a temporal logic, and model checking is defined as the study of all s states that are $s := f$ in M structure [34, 17]. The model checking is essential during the design phase since it detects the failures during the design stage [17]. Often it is very hard or nearly impossible to identify or rectify the failures after the design phase [17]. Celebi further mentions it is feasible to express the behaviour of the smaller system with a modelling technique such as petri nets, state-transition diagrams or ASM [17]. However, it is not practical to express the behaviour of a complex system due to the problem of state explosion [35, 17].

3.2.4 Graph Theory

Wang et al. presented a graph theory based approach to route location in railway interlocking [18]. A topology graph represented the station layout which was based on the component models. Thereafter graph theoretic methods were applied to evaluate the route information from the topology graph [18].

Each signalling element of the station was assigned with a component on the graph. These components were incorporated with a set of rules for the transformation of the railway signalling layout to component based topology graph [18]. Components mainly had two properties, which are vertex and edge. Vertex referred to the components of the graph and edge referred to the links between the components [18].

Graphs are presented through matrices. A set of matrices needs to be resolved as per the algorithm to obtain the route information [18]. For the computation of the route information a strict step-by-step complex processes were followed to determine the final matrix of route information. The final matrix produces all possible routes of the station layout [18].

One of the drawbacks of this method is that the algorithm is unable to evaluate the conflicting routes [18]. A separate algorithm needed to derive the conflicting routes. Wang et al. presented a purely mathematical exercise which did not show any coding and software configurations to auto generation of the control tables [18].

Wei Wong developed tools to produce control tables [36]. A graph theory was expressed in a Higher-Order Logic (HOL) [36]. A formal specification was produced by analysing the functional requirement of the system. Formal languages are used to capture the specifications of the functional requirements [36].

Wong further derived HOL theorems and HOL definitions to resolve the graphs in railway signalling applications [36]. The theories are developed in HOL to model the railway network components and functions of these components.

All paths are not legal routes. The algorithm produces all possible routes of the layout. Only then conflicting routes are eliminated from the route list [36]. Often these needed to be checked by an experienced signalling engineer [36].

The first task of generating the control tables is to represent all safety rules in codes. Safety rules needed to be defined in HOL codes [36]. The graph theory provides a sound mathematical structure as a base to evaluate the routes of the control table. The complicated railway tracks are intrinsically well suited to be represented by an abstract model of graph theory [36].

However it would be very difficult to model the railway network without a proper data structure. In this case, a formal method is used using formal language, HOL, to clearly define the data structure [36].

3.3 The Outcome of Previous Works

A number of research groups developed different techniques to automate the verification process of the control tables to reduce the computational time and memory usage on different size of station layouts ranging from smaller to larger stations. Several research groups gained success on verifying control tables for a smaller stations, but the current automatic verification techniques are challenged due to the explosion of state space size as soon as these techniques are applied on a larger station layout [7, 14, 19]. As the size of the station layout increase the computation time and memory usage of the control table increase exponentially [7, 14, 19].

H. Macedo et al. performed a great analysis on the use of compositional approach for verifying the interlocking system and produced some remarkable results. This research group devised a case study based on some of the common station layouts and evaluated the performance of the compositional approach on these layouts [14].

The case study includes three layouts, namely, Mini, Tiny and Fork. Mini refers to a single loop station, Tiny refers to a section of a single line and Fork refers to the layout configuration that starts with a single line converted to three lines which resembles a fork. These layouts are very common in signalling environments [14].

These layouts are then joined together horizontally to create a monolithic network. Thereafter these sub-layouts were verified separately using compositional approach. The larger monolithic layout was also verified. The running time and memory usage of the compositional approach were compared with the monolithic approach [14]. The monolithic approach verifies the entire monolithic layout at once. The table 3.1 shows the running time and memory usage by the verification tools [14].

Table 3.1: Verification tool running time and memory used by different approaches [14]

Approach	Time	Memory
Compositional Approach	29 sec	391 MB
Monolithic Approach	145 sec	759 MB

In compositional approach verification time was further reduced to 18 seconds by verifying the layouts in parallel [14]. The monolithic approach took five times slower than the compositional approach and took almost two times more memory than the

compositional approach [14].

Verification tools were then applied to a faulty model by introducing faults into the model and the performance of the both approaches were recorded [14]. The table 3.2 shows the running time and memory usage by the verification tools for a faulty model [14].

Table 3.2: Verification tool running time and memory used by different approaches for a faulty model[14]

Approach	Time	Memory
Compositional Approach	40 min	7.2 GB
Monolithic Approach	6.6 hours	20.7 GB

It is clearly evident for the results that the compositional verification tool becomes more efficient to verify the model with faults.

Compositional verification approach is a latest verification technique that has been exploited by the several railway focus groups. Most of the academic papers on the verification of the control tables using compositional approach are published around 2016 onward [7, 14]. Hence it can be inferred that compositional approach is fairly the latest technique within the signalling profession.

In compositional verification technique the entire large station layout is divided in smaller sub-sections and thereafter verified separately [7, 14]. The verification effort amount just to the sum of the verification effort for smaller sub-sections [14]. Research has shown using the compositional technique, the verification effort has reduced from exponential to linear computation time and memory usage.

3.4 The Focus Point of This Research

The auto generation of railway control table has been an active research topic for a few decades for the signalling research groups. Through all times there are various software tools developed for generating, editing and verifying the control table, there are still a few drawbacks and plenty of room for improvements [6]. Firstly, existing software tools are usually bound up with a specific railway company. Secondly, rules and regulations of railway companies differ from each other that control tables need

to comply with [6]. Thirdly and most importantly, automatically generated control tables are verified by signalling engineers manually which is very tedious, labour intensive and prone to make mistakes.

The latest attempt made by the research groups utilise the compositional approach to verify the control table. In best of the author's knowledge, currently researchers have applied this technique only to verify the control table. Traditionally control table is generated using all the elements of signalling layout linking them as per their geographical position. In compositional approach the layout is divided into smaller sub-sections and the control table is generated using these sub-sections.

It was perceived through this literature review that there has not been enough investigation on applying the compositional approach to generate the control tables. Therefore it opts for more investigation to be undertaken for generating control table using compositional approach. This research will focus on using the compositional approach to generate the control tables.

Chapter 4

Background

This chapter discusses background information associated with railway signalling discipline. A brief description is given on the history of the railway signalling. The chapter explains the purpose of the signalling in the railway environment. A complete elucidation of the major signalling equipment and functionality of these equipment are provided. Essential signalling rules and regulations are outlined in order to explain the safety critical train authorisation system. Finally, the fundamental principles of the control table for permissible train movement is illustrated in detail.

4.1 History of the Railway

The Reisszug, a funicular railway at the Hohensalzburg Fortress in Austria is believed to be the oldest railway which was built in 1515 [37]. Cardinal Matthäus Lang mentions that the Reisszug was originally made of wooden rails and a hemp haulage rope and this was operated by human or animal power through a tread wheel [37].The wooden railway tracks wore out quickly hence to overcome this shortcoming iron plated wooden rails were introduced. The Coalbrookdale Company began to install cast iron plates on the upper surface of the wooden rails in the late 1760s and this system was known as *Plateway* [37].

Clark mentioned the concept of railway has been initiated more than 250 years ago whereby the vehicle with metal wheels drove on a guideway metal rails [4]. The organised laid down wooden pathway for the wagons were used in mining industry where conditions were not suitable other mode of transport. The earliest record of

using wooden rails for coal mining is dated from 1630 in Britain which most likely the introduction of the railway transportation [4]. The first cast iron rails fastened to wooden sleepers were used in the Newcastle's Colliery in 1776 [4].

A British Inventor and mechanical engineer, Richard Trevithick, constructed the world's first full-scale high-pressured working railway steam locomotive in 1803 which he drove through streets of London [38]. The locomotive was first used for a railway by hauling a train with a load of 10 tons of Iron and 70 men along a 10 miles of tramway on 21st February 1804 [38].

4.2 The Origin of Signalling

There wasn't any early warning system available in the initial years of railways that could aware the train driver of the availability of the section ahead of him. Unlike the roads, the railway system's limitation is that the trains can only travel on the rails. In the event when the train steers off the rail, which is called a derailment, may cause a fatal accident. The train driver depended merely on his vision to avoid any obstruction on his line of sight on railway lines. The driver had to instantly make a correct decision of applying brake sufficiently early enough to prevent making an accident.

This method of operation is highly unreliable and susceptible of accidents. It was easily understood by the railway authorities that relying only on the driver's sighting distance to avoid accidents is not sufficient. This shortfall in railways has inclined to establish a new discipline called *Railway Signalling System* which involves the train authorising control methods.

Clark wrote about the Liverpool and Manchester Railway, the world's first passenger and goods train propelled by locomotives in September 1830 [4]. He described the ironic grand opening of the first passenger train turned into Britain's first public railway accident. Through this incident the local MP, William Huskisson, was run down by a train named Rocket which was unable to stop in time before hitting Huskisson [4]. This incident triggered the railway authority to start investigating on a development of a new railway discipline which is signalling. Clark marked this as the beginning of the *"Railway Signalling Age"* [4].

4.3 The Evolution of Signalling

In the beginning of the signalling era the entire signalling system was human based and the complete train authorisation system were controlled by signalmen. A set of hand signal rules were established to for train movement control. These signalmen carried out some of the most safety critical activities, such as, setting the route for the departing trains, ensuring the line clear for the arriving trains, and making sure points were lying in the correct position [39].

Fixed signals were located in the fixed position of the wayside of the tracks and comprised of movable flags, discs and coloured lamps which were manually operated by the signalmen [4]. These fixed signals began to replace the signalmen's hand signals in around the late 1830s [4]. This method of operation was purely based on time interval system which is letting one train within the section and thereafter waiting for a fixed period of time before letting the next train in the section [39].The downfall of this system was if the previous train broke down in the section and blocked the section then the approaching train had no knowledge of the blocked section which made the method of operation unreliable [39].

As time went by the railways were getting busier and the station layout were getting more complex hence there arose a need for a more reliable signalling system. Semaphore signals came in existence in 1841 and used to control train movements using mechanical interlocked lever frames in signal boxes [4]. Charles Hutton Gregory built an *Interlocking System* using a contraption of levers and stirrups in 1843, which operated a number of signals and points from a central location [4].

Once the electric telegraph was invented the railway authorities used it to communicate between the stations which was a major improvement from the time interval system. Signalmen were sending telegraphic messages to next stations confirming the arrival of train to the next station which means the section between the stations is free for the next train [39]. Thus the *Absolute Block Section* working was achieved. Absolute Block Section refers to method of working whereby only one train is permitted to travel at a time within the section [39].

Eventually the traditional mechanical interlocking was superseded by the relay-based interlocking system [1]. Electro-magnetic relays were used for complex interlocking electrical circuits which ensured the safety of the train authorisation system and prevented any possible conflicting train movements. Today's state-of-the-art signalling

system is the microprocessor based electronic interlocking system.

4.4 The Purpose of Signalling

The challenging part of driving a train is to stop the train. It requires a great deal of driving skill and concentration for bringing the train to a stationary position from a high speed [40]. The adhesion level of the metal wheels on the metal rails is much lower than the tyre on the tar roads which makes the railway transportation more energy efficient and environmental friendly than transportation via road [1]. The adhesion level between a tyre and the road is measured around 85% whereby only 8% adhesion level between the train and rail is measured for the breaking distance calculation of the train [40].

The train's weight on the railway is much higher than the vehicle's on the road. Trains requires considerably longer braking distance due to the low friction between the steel wheels of train and steel rail than the car with rubber tyres on the road. Therefore a train requires a much greater distance to stop on the rail compared to a vehicle of a same velocity on the road [1]. Train moves on a straight path on the rail and there is no opportunity for the train to steer off the rail without causing a derailment. If there is an obstruction within in the braking distance of the train then there is nothing that the driver can do to avoid that obstruction. Consequently the braking distance in front of the train must be free of obstruction for it to be able to stop in time safely.

The primary function of the signalling is to operate train safely. Nonetheless, signalling plays an essential role in increasing the capacity of the railway network besides satisfying the primary function of providing safety critical system. The increase in capacity of the railway refers to the increase in the throughput of the trains in the railway network.

Colin identified some of the high level responsibilities of the signal [3]:

- Ensure safe train route which is free from any human error
- Ensure to deliver the operational capability to execute the required services effectively
- Ensure the optimisation of the existing railway infrastructure usage safely

Colin further elaborated about the safe train route and delivering the operational capability. The following two fundamental rules must be adhered to provide the safe train route [3]:

- Reservation of a route which is protected from any human error, i.e. setting a safe route for the train
- Reservation of safe spacing between trains which is also protected from any human error, i.e. setting a safe overlap for the route of the train

The following four services shall be provided to deliver the operational capability [3]:

- Establish a safe environment for the safe train movement to achieve the service level requirement which must be fit for the site specific operational requirements
- Establish an adequate service providing management system and governance for signalling rail safety regulations
- Establish a proper communication system between staff and passengers
- Establish a proper asset management system which gives access to all stakeholders of railway for monitoring and maintaining the railway assets.

4.5 The Basic Elements of Signalling

The railway signalling elements can classified into two categories, for instance some equipment are trackside elements and others are central train authorising control elements. A brief description of the main signalling equipment are provided in section 4.5.1 - 4.5.7. These elements are identified with their assigned symbols in the signalling layout. The symbols that are used in this dissertation are usually used in South African Railway.

4.5.1 Signal

Signal is a signalling equipment that provides fail-safe instruction on how to operate the train to the train driver. Signal is placed by the wayside of the railway track

and it can only be seen from one direction of the track [12]. It is used to provide authority to a train driver to move onto the tracks ahead of the signal up to the next signal reading in the same direction [12]. Each signal has its own limit of authority on the railway. A symbol of signal is shown in the Figure 4.1.

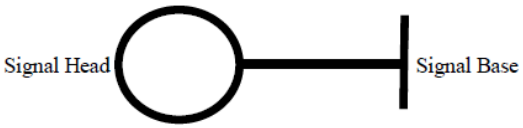


Figure 4.1: A symbol of a Signal

Drivers are expected to follow the instruction of the signals at all cost unless a railway official instruct the driver to supersede the instruction of the signal. Signals consist of different aspects to instruct the driver with different instructions. Aspect information is a method of indication which is used to display various information through to the train driver [41]. Eris and Mutlu listed the following information that can be given through the signal aspects [8]:

- Status of the tracks ahead, i.e. occupied tracks or vacant tracks.
- Availability of the route ahead
- Granting permission to the train driver from proceeding forward safely
- Preventing the train driver from proceeding forward
- Warning the train driver of the switch point in the route ahead, i.e. making the driver aware of the turnout
- Correct speed that train driver should maintain while travelling the section
- Possible distance to the next station
- Preparing the train driver for the possible state of the next signal

Railway organisation are responsible for specifying, managing and implementing aspects of the signal. Therefore it is the duty of the organisation to draft a set of unambiguous aspects information whereby the drivers are able to interpret the aspect clearly and instantly. The driver of a particular section need to be trained to be able

to drive on that section so that the driver is familiar with the signal's aspects of that section and able to correctly interpret the signal's aspects.

The aspect information of the signals are not universal and it varies between railway authorities. Different regions or countries may employ different method to display aspects differently [41]. The three-aspect signals are the most common aspect displaying method used worldwide. Red, Yellow and Green are the aspects of three-aspect method. The Table 4.1 shows the aspect meaning of the three-aspect signal:

Table 4.1: The symbols of the signal aspects

Aspect	Symbol	Meaning
		Proceed and next signal is displaying a proceed aspect and there are no turnouts in the route or overlap ahead
		Proceed and be prepared to stop at next signal unless it displaying a proceed aspect and this aspect also indicates that the driver must expect to deviate from straight route or overlap ahead over a low speed point set
		Danger and the driver must stop in front of the signal
		Indicates the driver Right Routes or Left Routes as per the route indicator

There are other aspect displaying method, such as, two–aspect, four-aspect signal or multi-aspect signal. The two-aspect signals are used in low speed lines where as the four-aspect signals are used in high speed lines [1]. Multi-aspect signal makes allowance for displaying a combination of aspect simultaneously. In this research work, the method of three-aspect signal will be used since it is widely used throughout the world.

4.5.2 Point

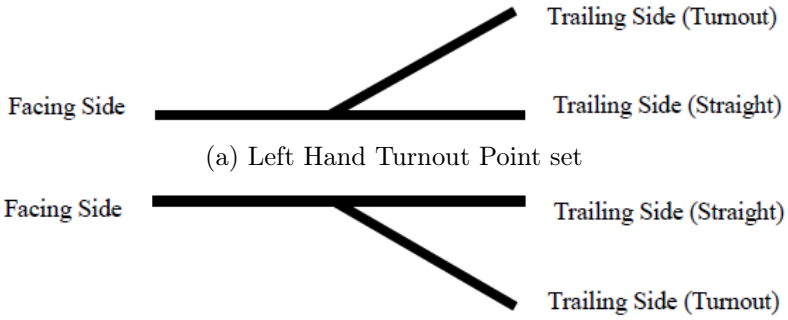
Point set is a special arrangement of track section which allows of merging two railway lines into one railway line [12]. Point is a mechanical equipment usually driven by electrical motor that enables train to change their direction or route from one track to another [41]. There are other types of point's driving mechanism available, such as manually operated hand tumbler, trailable point set, etc.

Railway vehicles do not need to be equipped with a steering mechanism in the vehicle since these vehicles are only able to move on the guide way of the rails [13]. Therefore point sets are very critical rail track equipment as without the point sets train will not be able to move from one line to another. The point set generally have two positions, namely straight - which is also known as normal - and turnout - which are also known as reverse [12].

Points must be installed with the locking mechanism and the detection mechanism to able to lock and detect the point for a known definite position of the point [13]. Trains are only permitted to traverse over a point set when the point set is locked for a known position either straight or turnout due to the high safety requirement.

On a Left Hand Turnout the straight position is found on the right of the point and the turnout position is found on the left of the point. A symbol of the Left Hand Turnout point set is given in Figure 4.2a.

On a Right Hand Turnout point the straight position is found on the left of the point and the turnout position is found on the right of the point. A symbol of the Right Hand Turnout point set is given in Figure 4.2b.

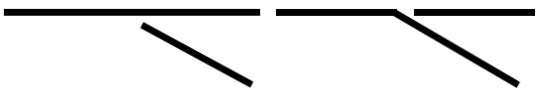


(a) Left Hand Turnout Point set

(b) Right Hand Turnout Point set

Figure 4.2: The Symbols of the point sets

The default position of the point is shown Figure 4.3.



(a) Straight position (b) Turnout position

Figure 4.3: Symbol of the default position of the point set

4.5.3 Derailer

Derailer is a mechanical device which is used to derail the runaway wagon to protect the running main line movements. Therefore the main function of the derailer is to prevent collisions by purposely derailing unauthorised movements or deviating them away from the authorised movement.

The uncoupled empty wagons are often parked without locomotives within the sidings or in the yard. In the event when the brakes of the wagons do not function properly and concurrently due to the steep gradient of the yard or siding, the wagons start moving downward. These runaway wagons are the cause of an accident waiting to happen. Therefore these runaway wagons are purposely derailed from the rail to protect the mainline trains [42].

The positioning of the derailer must be carefully considered and the most suitable and safe location must be chosen for the installation of derailers. The space between the tracks must be considered while selecting the location of the derailer because the derailed wagons must not derail onto the adjacent railway line. Place a sand bank or a ballast drag at the edge of the derailer to decrease the momentum of the derailed wagons [42].

Sometimes the wagons may jump over the derailer and continue moving on the track due to high momentum and the heavy weight of wagons. In that case a point set should be used as the derailer to derail the wagons which is also known as runaway point set.

Figure 4.4: A symbol of a Derailer

4.5.4 Train Detection

Train detection equipment is a device that able to detect the presence or absence of a train on a railway track in a fail-safe mode [13]. A long section of track is divided into segments of tracks and each segment is equipped with a train detection equipment that detect the presence of train on the segment [12]. This device is used to inform the Traffic Control Officer (TCO) about the status of the track, namely the occupancy

or the vacancy of the track. The following two types of train detectors are the most common in railway:

1. **Track Circuit**

   Track circuit is an electrical device that locates the position of the train once it enters the segment of track where this device is installed. A segment of the rail of a track circuit are insulated from the rest of the track using block joints. The current flows through both rails energise a track relay to ensure that track is free of any vehicle. In the event when a train arrives within the segment of the track which is equipped with track circuit, a new path of current flow is created through the wheels and the axles of the train with much low resistance which consequently de-energise the track relay to ensure that track is occupied with a vehicle. However there are different types of track circuits, for example, jointless track circuit which mainly works with frequency of signals from the transmitter and receiver of the track circuit.

Figure 4.5: A symbol of a Track Circuit

2. **Axle Counter**

   Axle counter is also a train detection device that counts the number of axles of the train for a section of the track. The section of the track is installed with a least two or more wheel sensors which is able to count the axles of train. Axles are counted in while entering into the section and also axles are counter out while exiting the section. By counting in and out of the section, the completeness of train in determined within the section. Thus the presence of the train is determined within the section.

Figure 4.6: A symbol of an Axle Counter Track

4.5.5 Remote Control System

The remote control system allows an operator to control signalling equipment from the Centralised Traffic Control (CTC) which is located far away from the actual

railway network. It operates various critical train authorisation activities, such as, operating the points, setting the routes and clearing the signals, etc. The remote control system must make use of a visual unit which enables the operator to visually observe the real-time true state of the railway stations. The control system can be divided into two parts, namely, transmission and reception. Transmission sends control instruction from the CTC to the equipment on the ground via relay rooms or equipment rooms. Reception collects indications from the equipment via remote relay rooms or equipment rooms.

4.5.6 Interlocking System

The interlocking is the core component of the signalling system. Interlocking ensures all signalling elements of the station are interlocked with each other for safe operation of train movement. Signals, points, track circuits are interlocked to ensure that no dangerous or conflicting train movements may occur [43]. Interlocking controls and monitors all the routes of the stations which are interlocked for secured route reservation of the train.

Signalling rules and regulations strictly describe what must be done and what must not be done during train authorisation. It is fundamentally important to follow these rules and regulations completely and exactly to avoid any accident on the railway. Interlocking is the hardware and software device that confirms these rules and regulations are adhered entirely for train authorisation. The main functions of the interlocking are as follows:

- When a route is requested by the TCO then the interlocking must verify that the entire route from the departure signal to the destination signal is available [1],
- When a route is assigned for a train then the interlocking must ensure that route is exclusively reserved for the allocated train only and this route cannot be shared with any other train [43],
- Interlocking must provide protection against human error by confirming any conflicting routes are not be set up by the TCO intentionally or unintentionally[3],

- Opposing signals must never be able to operate simultaneously to avoid an accident [43],
- Interlocking must check the status of the tracks continuously at real-time and only reserve the tracks for the route if it is vacant [1],
- All set of points within the route must be detected and locked in the corrected position prior to display proceed aspect on a signal [43],
- Once the route is locked with all the elements, such as points, tracks, signals of this route are reserved only then signal may display proceed aspect [1],
- Once the train enters the route then the departure signal must immediately fall back to danger to prevent the subsequent trains entering into the route which may cause a rear-end collision [1],
- Making sure of the optimum use of the railway capacity with maximum number of trains on the line can operate safely using safe train spacing [3].

4.5.7 Other Equipment

There are various other signalling equipment that are often used within the railway network. The list below describe some of the equipment that are widely used within signalling environment:

1. **Crank Handle**

   Crank handle is often used to throw a set of point manually when the electrical power of point's set is unavailable.

2. **Key Siding**

   When a train needs to entre into a siding from the running mainline which is protected by a key unit, then the driver needs to obtain the key from that unit. However the key is electrically protected by the interlocking and it can only be released remotely by the CTC operator. Once the CTC operator releases the key remotely then the driver can physically obtain the key that will allow the driver to enter into the siding.

3. **Siren**

   Siren is often used within the railway station to alarm the maintenance workers about the approaching train to the station.

4. **Hot Box Detector**

   Hot box detector detects any axle that has excessively high temperature and may cause an accident in the network.

This equipment are equally important for the railway and performs very critical functions. There are many other units of equipment that used to carry out various other functions throughout the railways. However, this research work is concentrated around the basic equipment – signal, point, and track – which are the most important equipment and carries out the most crucial tasks. The picture in Figure 4.7 in shows a high level interaction diagram between the basic equipment.

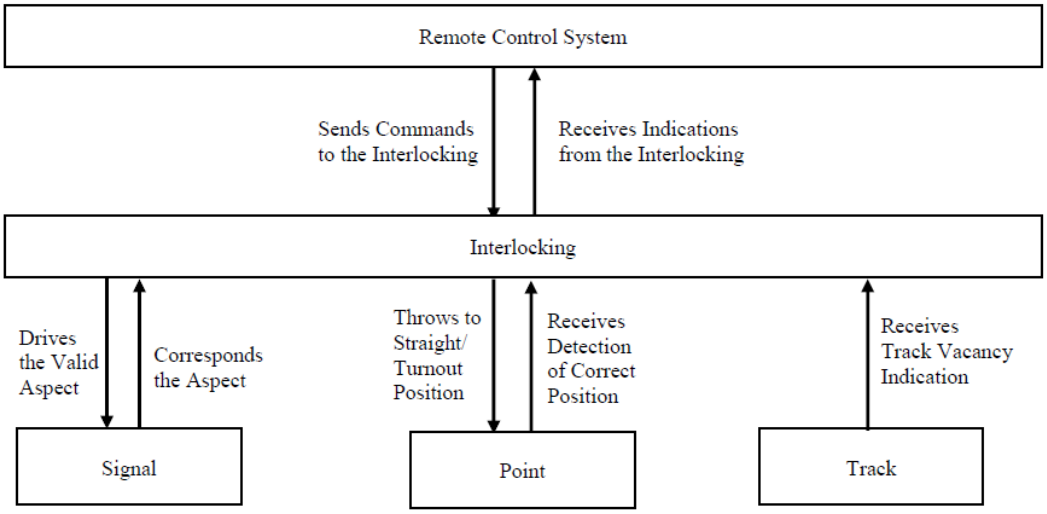


Figure 4.7: Interaction between the Signalling Elements

4.6 Signalling Principles

Signalling system executes the safety critical task and it requires the safest working conditions in railway. Therefore the main purpose of the railway signalling discipline is to ensure the safest train authorisation takes place within the railway network. Safe operation of train movement includes the most safety critical functions such as, avoid conflicting movement of trains, prevent derailment of rail vehicle, and prevent collision over a level crossing with the road vehicle, provide safe operating distance between trains, etc.

The responsibility of the signalling system is to implement a fail-safe communication between the TCO and the Train driver which enables a safe and efficient train movement in the railway network. TCO is stationed in the CTC whereby TCO receives the real-time indication of the trackside equipment and the train occupancy status of the railway. TCO is able to send the command to control the trackside equipment remotely from centralised control station.

This dissertation lists some of the most critical signalling safe working rules. It is standard practice within the South African Railway industries to regularly apply these safe working rules. The signalling systems must incorporate the following cardinal safety rules of signalling:

Rule 1: Fail-safe Equipment

A fail-safe system refers to the system which is able to fall-back into a safe state and continue operate safely in the event of a failure. Signalling system cannot afford to make any mistakes while authorising the trains because any mistake may cause devastating disaster. Therefore all signalling equipment are safety critical elements of the system and must be fail-safe equipment. The primary functions of the fail-safe system can be described as follows [44]:

- Ensure to drive the system safely after an occurrence of a fault
- system must operate into a predefined safe state in the event of a failure

Train authorisation is a safety critical task and a great importance must be given to the design and development of this system. A component of safety critical system must perform a right side failure. For instance the right side failure can be described when a signal fails it must fall to a danger aspect and must be prevented to display a proceed aspect. If the proceed aspect is displayed in the faulty signal then it would be a wrong side failure. It is critical for the fail-safe system to fail in right side failure unless this could result in a critical accident.

Rule 2: Setting of the Route

Intended route must be set for the train from the departure signal to the destination signal. Employed system must ensure that all the element between the departure signal to destination signal are available to use. If an element is already occupied or unavailable then route cannot be set over that element. For example, if a point set is under maintenance and the operator sets a route over that point set then it will cause an accident over the point set.

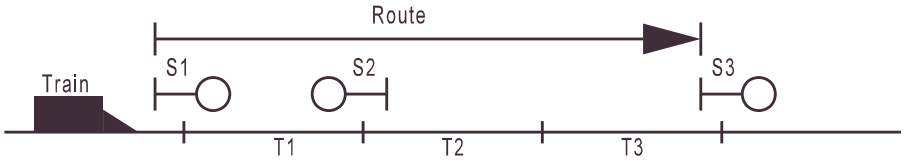


Figure 4.8: A route

In Figure 4.8 illustrates a route that extends from S1 to S3. In the above route, S1 is the Departure signal, S2 is the opposing signal and S3 is the destination signal. T1, T2 and T3 are the tracks in the route. For this to be a valid route for a train the T1, T2 and T3 tracks must be vacant. The opposing signal S2 must be at danger so that the route is protected from the opposite direction. The departure signal S1 must display a proceed aspect and S3 shall display danger for the train to stop short of the signal. The route in Figure 4.8 does not contain any point set.

Rule 3: Lock and Detect the Points

Once the route is set then all the position of points has to be detected and confirmed that the position of the point is correct for the route. When the position of the point is detected correctly and verified then the point set must be locked for the desired position. This means the point set shall not change the position while the route is set for the train. These point sets are equipped with mechanical locking device at the point's machine and over and above that the point sets are also electrically locked from the interlocking.

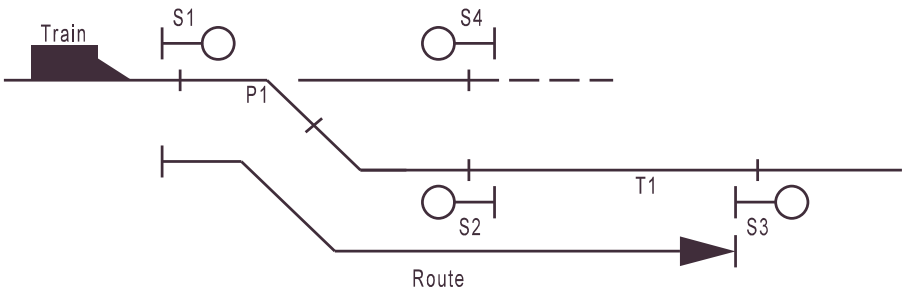


Figure 4.9: Point detected and locked

The route in Figure 4.9 includes a point P1 which shall be set in the turnout position. For the route from S1 to S3 to be valid, the point P1 must be locked and detected in the turnout position. The turnout position of the point P1 must be proved in setting of this route. If P1 losses its detection then the route shall no longer be available for the train.

Rule 4: Lock the Direction of the Track

Firstly the tracks shall be unoccupied and available for the route. If the tracks are free then a valid route may be called over the tracks. These tracks also need to be locked for the route with direction. In other words if the route is called from left to right of the track then track must be locked with left to right direction.

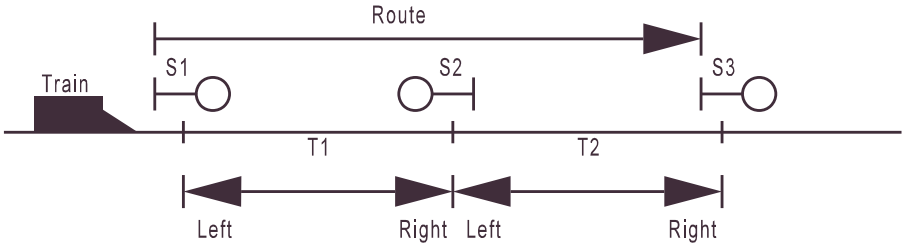


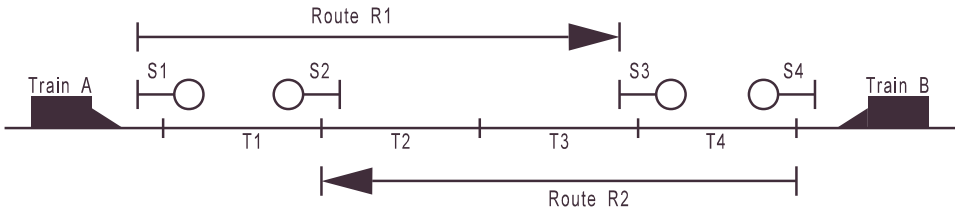
Figure 4.10: Lock the direction of the track

Since the route is set from left to right as shown in Figure 4.10 hence the train is expected to move from the signal S1 to the signal S3. In this case, interlocking system is expecting a movement from Left of T1 to right of the T1 and similar movement for the track T2 as well. In order to ensure the correct sequential movement of the track T1 must be occupied first and only then track T2 may be occupied before it reaches to the signal S3.

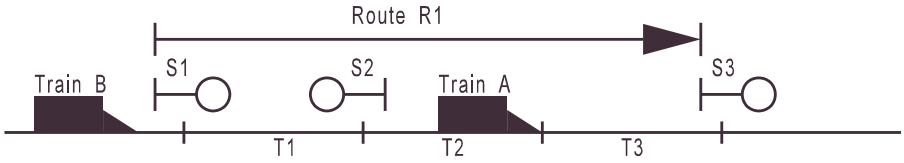
Rule 5: Prevent Conflicting Movements

The fundamental principal of signalling system is to avoid making use of an element for more than one route. A route or a portion of the route shall not be shared by two trains. Conflicting movement may occur in different ways, for example:

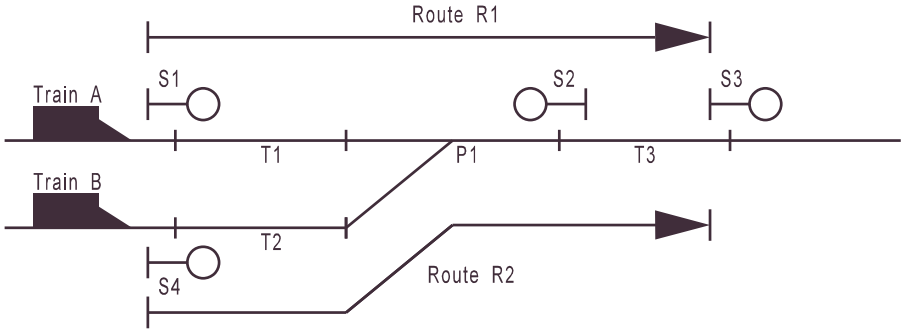
- If two routes were called face to face of each other sharing the elements then that will cause a head-on collision.
- If a train moves into a route and subsequent train moves into the same route from the same direction then that will cause a rear-end collision.
- If two train meets each other over the trailing side of a point set that then will cause a side collision.



(a) Conflicting movement that can cause a head-on collision



(b) Conflicting movement that can cause a rear-end collision



(c) Conflicting movement that can cause a side collision

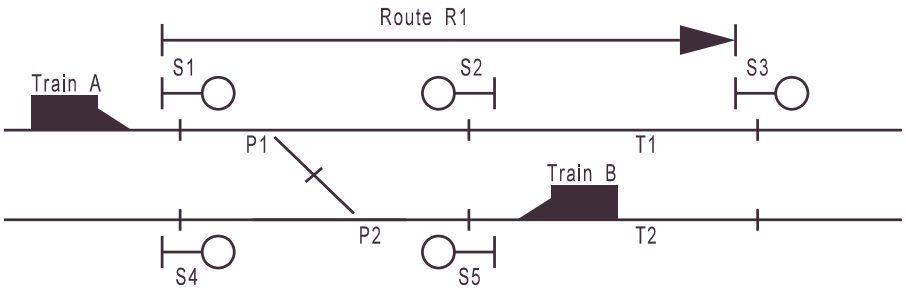
Figure 4.11: Examples of conflicting movements

Figure 4.11 depicts three examples of conflicting movement on the railway track. In Figure 4.11a shows two conflicting routes, namely R1 and R2, are set that will cause a head-on collision between the train A and B. Figure 4.11b shows while there is 'Train A' within the route R1 but the route R1 is set for the 'Train B' which will cause a read-end collision. In Figure 4.11c shows the route R1 and R2 is set over the same point and may cause a collision between train A and B over the point set P1. Signalling system shall be designed such that it can prevent all different types of conflicting movements.

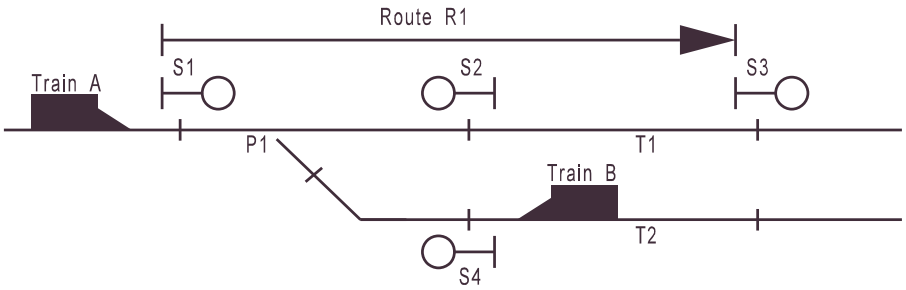
Rule 6: Provide Flank Protection

Whenever there are point sets within the route then the route must be protected by flank protection. The flank protection protects the route from the side. Points has two trailing sides, if a route is set over straight position of the point then the route shall be protected from the turnout side of the point. System must ensure no train is able to collide with this train coming from the turnout position. In other word, route is set over one mainline must get flank protection from other lines.

The first preference of obtaining the flank protection is from the point set. Should the point sets are unable to provide flank or there are no point set to provide flank protection, in that case flank protection should come from signal. These are the main two element that provide flank protection. However, sometimes even tracks can also provide flank protection for the complex layout. In this research work, flank protection from points and signals will be considered.



(a) Point providing flank protection



(b) Signal providing flank protection

Figure 4.12: Examples of flank protection

Figure 4.12 represents two examples of flank protection. In Figure 4.12a shows the flank protection obtained through point set. The Route R1 set over one mainline which contains the point set P1. For the valid route R1 the point P1 shall be positioned for straight. Moreover the point P1 leads to the point set P2 where P2 is able to provide flank protection to R1 by making sure P2 is also set for straight position. Thus the train B will not be able to collide with train A.

Nevertheless in Figure 4.12b shows there are no point set to provide flank protection. In that case the signal S4 needs to provide flank protection to the route R1. The signal S4 must display Red to warn the driver of train B so that the driver does not move pass the signal S4.

Rule 7: Provide Overlap of the Route

Overlap is an extra distance of clearance provided to the route after the destination signal in case the driver does not apply brake on time and passes the destination signal. Every route shall be provided with an overlap. There are various overlap measurement used by the railway authorities throughout the world.

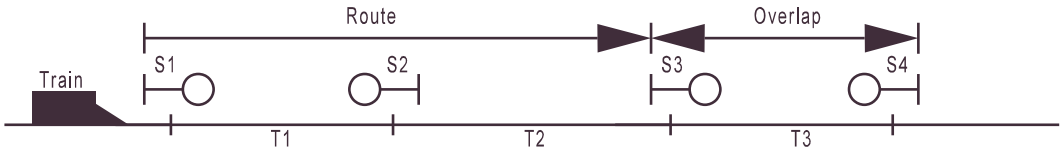


Figure 4.13: Overlap

Figure 4.13 provides an depiction of the overlap of the route which extends from S3 to S4.

Rule 8: Provide Head Protection

The head protection avoids the head-on collision. When the signalling system is designed with head protection then the trains cannot collide head-to-head. If there is a point set installed pass the destination signal and the overlap distance then the point set must be thrown for the opposite position of the route's direction.

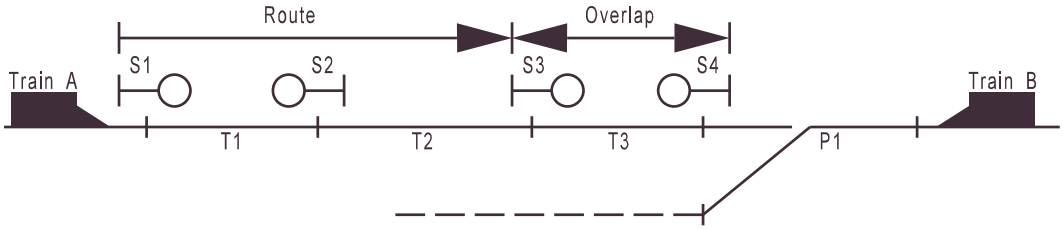


Figure 4.14: Head Protection

An illustration is made of head protection in Figure 4.14 that illustrates point P1 provides head protection to the route. The point set P1 sets for the turnout position which will ensure the train B may not move into the boundary of the overlap and route.

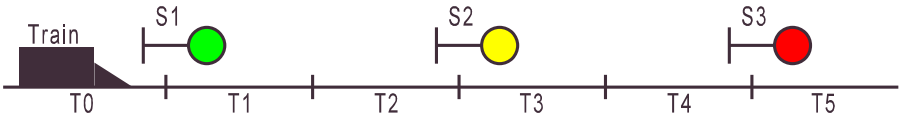
Rule 9: Aspect Display

A train just does not stop immediately in fact it takes a long distance to come to a halt. Even with the best braking system, due to the low friction between the metal wheels on the metal rail require a much longer distance to stop the train.

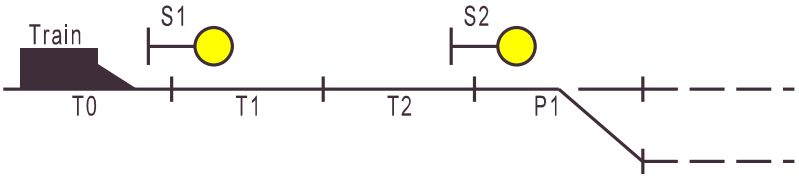
The train driver needs to be warned much earlier so that driver is able to slow down and be prepared to stop ahead of the destination signal. Train driver also needs to be informed prior to coming closer to the turnout.

Yellow aspect serves as a warning signal which warns the driver to stop in front of the next signal ahead. The picture below illustrates aspect ruling for the three aspect signalling

As mentioned in section 4.5.1, it is totally depended on the railway authority to what aspect displaying method shall be used on the rail. But the examples below are most common in the signalling industry and has to be satisfied by the signalling system.



(a) Train needs to gradually slow down to stop at danger



(b) Train needs to slow down to traverse over a point

Figure 4.15: Two basic examples of aspect ruling

Figure 4.15a shows the train cannot immediately come from the green aspect to stop at red aspect. A yellow aspect must be shown to warn the driver to slow down before coming towards a red aspect.

Similarly the train cannot move at the line speed over a point set with sharp turnout. In Figure 4.15b shows only yellow aspect on S2 is not sufficient for the train to turn on the point P1. In fact it's too late for the driver to slow the train enough to turn on the point P1 which may cause derailment. In this case, driver needs to be notified two signals earlier before reaching the turnout to properly handle the train on the turnout. Signal S1 needs to display yellow for driver to slow the train to properly traverse the train over the point set P1.

Rule 10: Sequential Route Release

Sequential route release must include the following:

- When a route was set and proceed aspect is displayed. Once a train goes into the route and it passes the departure signal, thereafter clears the berth track

of the departure signal then the departure signal must fall back to danger and shall not keep displaying the proceed aspect. This will cause a confusion for a subsequent train, which might move into the route that was not intended for it. Therefore, no sooner the first train clears the berth track the departure signal goes to danger which ensures another train may not follow the route mistakenly and cause a rear-end collision

- While train traverse over the track, then the tracks must be released sequentially, i.e, as train occupies the track and moves to the next track, interlocking must confirm that the next track is occupied before making the previous track vacant. If this sequence of occupied and unoccupied does not occur correctly then it must be understood train movement did not occur and there is a fault in the system
- During the travelling time of the train, the route must be kept locked for the train. Once the train traverses over the route sequentially and reaches the destination signal safely then after a short delay the route can be released. This is also known as automatic train normalisation.

Rule 11: Avoid Going Around the Bend in Double Crossover

This rule is entirely apply for the layout with a *Double Crossover* setup. Double crossover is often used between two mainlines. Double crossover refers to four point sets configured such a way that train from one mainline can move to the other main line. In Figure 4.16 shows an example of double crossover. There are more than one route to get to the destination signal as shown in Figure 4.16. However in this instance the straight route, Route R1, should be the default route. The interlocking must avoid setting the slightly longer route, Route R2, that goes around the bend in double crossover. This is to protect the train from breaking the coupler between two wagons. Couplers are used to join the wagons to make the complete train.

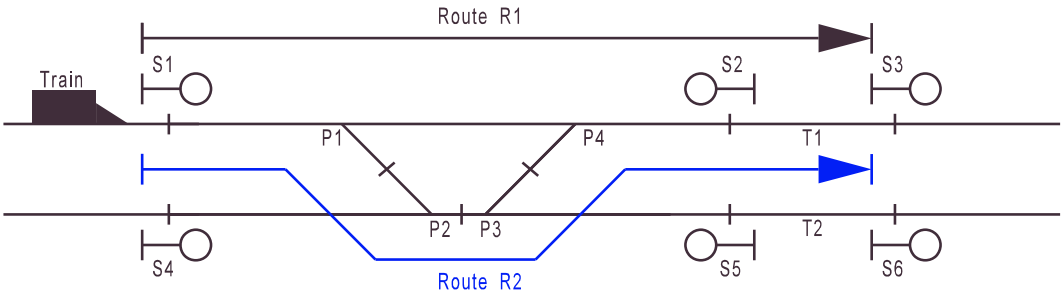


Figure 4.16: Select the straight route in the Double Crossover

It is nearly impossible to discuss all the signalling rules at length for this research work. Hence the most basic critical safety related rules are expounded and elucidated upon using pictures.

4.7 Control Table

The control table is the functional requirement for the railway signalling interlocking which plays a key role in train authorisation system [45]. The control table specifies all valid routes within a station on which trains are permitted to travel on. The control table adheres to the standard safety principle of signalling and fail-safe conditions of an interlocking system.

The control table is presented in a table format whereby all the fail-safe conditions of all possible routes of the train are provided [6]. The route of the train includes the path of train where the train will start the journey and end the journey and all required information such as the tracks and the points for the train to pass through safely as stated in the control table.

A control table is a structured and tabular presentation of all the required conditions for the valid train movement on a railway track layout [6]. The method of writing control tables are derived from the principles of safe train working rules [6]. A control table represents an intermediate level of design between the station track layout and interlocking logic circuits [1]. Kirsten Winter mentions three important purpose of the control table are[45]:

- It serves as an agreement between the railway administration and the train operator to move train on the railway tracks over that particular station.
- It provides a design specification of the interlocking.
- It acts as a test specification for test engineers.

Railway authorities throughout the world apply diverse safe train working rules. This is mainly due to, the geographical locations and conditions of the railways and the types of signalling system which is used within the railway network, that influence the railway governing bodies to apply diverse safe working practices [6, 1]. Hence the railway authorities tailor the structure and content of the control table to suit

their needs. Though the general principles of the control table are same across all the railway authorities, there is however no standard format or content of control table [6, 1].

There are mainly three sections of a control table, they are: route information, flank protection information and aspect information. The following columns of the control table provides the information in an organised manner. These columns are generated automatically in this research. The columns are outlined below:

- Route – all possible routes of the station are evaluated and each route is provided with its own number
- Departure signal – it is the start of the route
- Destination signal – it is the end of the route
- Track required – all the tracks that are required from the start of the journey till the end of the route
- Points required in straight position – points that must be set straight position for the route
- Point required in turnout position - points that must be set turnout position for the route
- Flank by point required in straight position - points that must set for straight position to provide flank protection for the route
- Flank by point required in turnout position - points that must set for turnout position to provide flank protection for the route
- Flank by signal - signals that need to set for danger to provide flank protection for the route
- Conditions for GREEN - aspects that are required on destination signal for the departure signal to display the GREEN aspect
- Conditions for YELLOW - aspects that are required on destination signal for the departure signal to display the YELLOW aspect

Figure 4.17 shows a station layout and Table 4.2 is the control table of this layout. The numbering of the signalling elements in this layout follows the standard numbering convention of South African Railways.

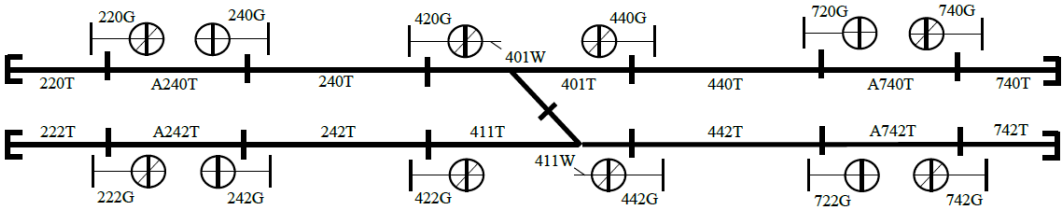


Figure 4.17: A station layout

Table 4.2: The control table

Route	Departure Signal	Destination Signal	Departure Signal Aspect		Tracks Required	Points in Route		Flank Protection		
			Requirements for GREEN	Requirements for YELLOW		Straight Position	Turnout Position	Points in Straight Position	Points in Turnout Position	Signal
R 1	220G	420G	420G GREEN, 420G YELLOW	420G GREEN, 420G YELLOW, 420G RED	A240T, 240T					
R 2	222G	422G	422G GREEN, 422G YELLOW	422G GREEN, 422G YELLOW, 422G RED	A242T, 242T					
R 3	420G	720G	720G GREEN	720G GREEN, 720G RED	401T, 440T	401W		411W		
R 4	420G	722G		722G GREEN, 722G RED	401T, 411T, 442T		401W, 411W			440G, 422G
R 5	440G	240G	240G GREEN	240G GREEN, 240G RED	401T, 240T	401W		411W		
R 6	422G	722G	722G GREEN	722G GREEN, 722G RED	411T, 442T	411W		401W		
R 7	442G	242G	242G GREEN	242G GREEN, 242G RED	411T, 242T	411W		401W		
R 8	442G	240G		240G GREEN, 240G RED	411T, 401T, 240T		411W, 401W			422G, 440G
R 9	740G	440G	440G GREEN, 440G YELLOW	440G GREEN, 440G YELLOW, 440G RED	A740T, 440T					
R 10	742G	442G	442G GREEN, 442G YELLOW	442G GREEN, 442G YELLOW, 442G RED	A742T, 442T					

Chapter 5

Research Methodology

The proposed methodologies for automatically generating the control table are discussed in this chapter. These methodologies includes conventional approach and compositional approach for interlocking control table generation using a software program. This chapter clearly elucidates the processes of software development. It further elaborates the crux of the various software techniques that was used to develop the program.

5.1 Modus Operandi

Research incorporates two components, these are, attempting different signalling methods for the generation of control table and the developing software program for these methods.

Signalling work of the research entails exploring the methods of generating control table in details. The research utilises two approaches to produce the control table, namely, conventional approach and compositional approach. Conventional approach has been used by various research groups and compositional approach is the latest technique within signalling profession.

Software development work of the research entails preparing the programs that are able to generate control tables using the above mentioned two approaches. Two algorithms were developed utilising conventional approach and compositional approach to generate the control table. Similar software techniques were applied to develop

the algorithms of the program to ensure a fair comparison can be made between the algorithms. The efficiency of the algorithms are compared in Chapter 6.

5.2 Software Development

A software development platform and a programming language is essential to develop the automatic control table generating program. The .Net framework is a software development platform for building and running applications that also supports various programming languages such as C#, F# and Visual Basic [46]. C# is the high-level programming language that was developed by Microsoft along with the .NET environment [47]. C# is a general purpose object oriented programming language and .NET is the runtime support environment [47]. Object Oriented Programming (OOP) is one of the most popular programming technique. C# also supports other features including abstraction, encapsulation, inheritance and polymorphism. Useful features such as inheritance, polymorphism came in handy for linking signalling elements in the software program. The automatic control table generating program is developed using C# language in .Net developer platform.

5.2.1 Object Oriented Programming (OOP)

Generally the purpose of the programming was to input data into a program and the program would execute a logical task to generate expected output data [48]. Primarily programs contained a simple structure such that a long list of instructions were executed as per the objective of the program [48]. The reusability of the portion of the program was not achievable easily. In order to overcome some of the challenges during the initial technique of programming the OOP was introduced.

OOP refers to the software development technique whereby the structure of the program is based on objects [49]. These programming objects send and receive instructions from each other to perform a certain task [49]. The objects consist of data and only programming default processes need to be followed to obtain or modify the data [48]. Using this programming technique the signalling elements of the station layout was defined as the software objects in the program.

Classes are the templates of the objects which are used to create the objects in the program [48]. Classes define the properties and methods of the objects [48]. Moreover,

the inheritance classes refer to those classes where the derived classes inherit the properties and methods from the base class. For the research project various classes are defined that were required to create the software objects of the signalling elements. For instance a base class was defined as the *Element* class and thereafter the derived classes were defined as *Signal*, *Point* and *Track*. Thus the inheritance technique of OOP is applied. The base class and derived class relationship is shown below in Figure 5.1.

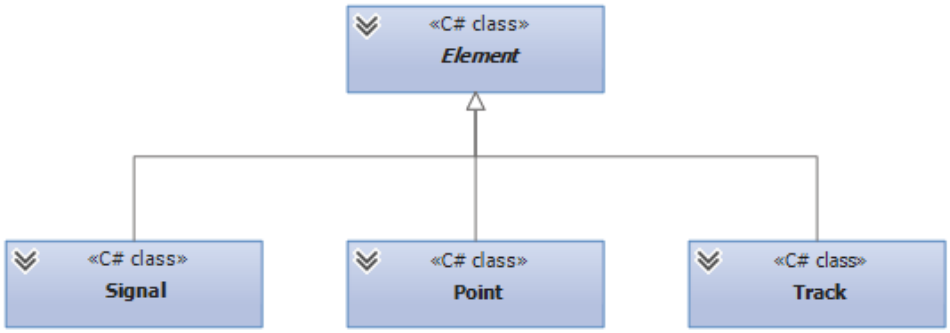


Figure 5.1: The inheritance diagram of the program

There are many other relationships between the classes that needed to established for successful interaction between the classes. The program frequently makes use of other class relationships such as, Association, Aggregation and Composition. Appendix B describes the classes and the relationships between the classes of the program in details.

5.2.2 Polymorphic Technique

Polymorphism is one of the most exceptional technique of OOP and often very complex to grasp the concept. The unique characteristic of polymorphism is that two different objects are able to respond to the same request in their own unique way [49]. For example, objects are able to invoke each other's methods [48].

In order to reap the benefits of polymorphism in OOP the method requires to be override-able. Methods of the different classes can be implemented using the same name with *Virtual* keyword. In other words the *virtual* methods of base case can

be overridden by the derive class. However, these methods of the derive class need to be *Override* methods. In the case of the research project, *Virtual* methods were implemented for the base class, *Element*. As per the polymorphic process, the *Override* methods were to be implemented for the derived class, such as, *Signal*, *Point*, *Track* and etc. The code in listing 5.1 provides the entire *public* interface of the *Element* parent class that has three child classes which are *Signal*, *Track* and the *Point* classes.

Listing 5.1: Snippet of C# code

```
1  public abstract class Element
2  {
3      public string ElementName { get; set; }
4      public virtual Element Right { get; set; }
5      public virtual Element Left { get; set; }
6      public virtual Element Base { get; set; }
7      public virtual Element Head { get; set; }
8      public virtual Element Straight { get; set; }
9      public virtual Element Turnout { get; set; }
10     public virtual Element Face { get; set; }
11     public virtual string PointTrack { get; set; }
12     public virtual SignalAspect[] Signal_Aspects { get; set; }
13 }
```

The *Element* class defines all the *set* and *get* methods. The *ElementName* is a *string* type property that stores the name of the element irrespective of the type of the element it is, example signal, point or track. It also defines the linking properties, such as, *Right*, *Left*, *Base, Head*, *Straight*, *Turnout* and *Face*. These linking properties are all *Virtual* methods. The *Element* class has another two methods. The method *PointTrack* only applies to the point class and *Signal_ Aspects* applies to signal class only. The *PointTrack* is a *string* type that stores the name of the point's track. The *Signal_ Aspects* stores an array of signal aspects. The code in listing 5.2, 5.3 and 5.4 shows the entire class implementation of *Track*, *Signal* and *Point* respectively.

Listing 5.2: Snippet of C# code

```
1  public class Track:Element
2  {
3      public Track(string name)
4      {
5          ElementName = name;
6      }
7      public override Element Right { get; set; }
8      public override Element Left { get; set; }
9  }
```

Listing 5.3: Snippet of C# code

```
1  public class Signal:Element
2  {
3      public Signal(string name, SignalAspect[] sig_aspts)
4      {
5          ElementName = name;
6          Signal_Aspects = sig_aspts;
7      }
8      public override Element Base { get; set; }
9      public override Element Head { get; set; }
10     public override SignalAspect[] Signal_Aspects { get; set; }
11 }
```

Listing 5.4: Snippet of C# code

```
1  public class Point:Element
2  {
3      public Point (string name, string ptrack)
4      {
5          ElementName = name;
6          PointTrack = ptrack;
7      }
8      public override string PointTrack { get; set; }
9      public override Element Straight { get; set; }
10     public override Element Turnout { get; set; }
11     public override Element Face { get; set; }
12 }
```

The *Track* class has a constructor method and two linking properties, these are *Right* and *Left*. These linking properties have the same signature as the *Element* class *Virtual* Methods. Hence these linking properties are overridable. Similarly, the properties of Signal and Track are overridable as well.

Initially a software program does not have any knowledge of the signalling station layout. Each signalling layout of the station will differ from each other. While defining the *Element* class, the challenge was to make the *Element* class able to link with other element classes without knowing which element that it might be. To overcome this challenge the *Element* class implemented *Virtual* linking methods. Once the elements of station layout were defined in the program, only then these elements are able to link with each other using the *Override* linking methods of derived class objects, such as, *Signals*, *Points*, *Tracks* and etc.

This unique ability of polymorphism made it possible to link the elements of the station with much more ease for the program. It would become very difficult to

implement this program without applying the polymorphic feature of OOP.

5.2.3 Object Linking Technique

Signal, *Point* and *Track* are child classes of the *Element*, base class. But these classes contain *Element* type linking properties which refer to the other elements. Figure 5.14 shows a small layout that has a signal, a point set and two tracks. The code in listing 5.5 shows how a signal, a point and two track objects are declared in the program and inserted in the respective *Lists* of *signals*, *points* and *tracks*.

Listing 5.5: Snippet of C# code

```
1 points.Add(new Point("P1W","P1T"));
2
3 SignalAspect[] Sig_S1G_Aspt = { SignalAspect.GREEN, SignalAspect.YELLOW,
      SignalAspect.RED };
4 signals.Add(new Signal("S1G", Sig_S1G_Aspt));
5
6 tracks.Add(new Track("T1T"));
7 tracks.Add(new Track("T2T"));
```

As it is shown in the listing 5.3 of *Signal* class that it has two linking properties, *Base* and *Head*, which are an *Element* type, therefore it is able to initialise to any of the other elements. In this case, *Base* was initialised to *null* and *Head* was initialised to *Point*, P1, as per the geographical position of the layout shown in Figure 5.14. The portion of code in listing 5.6 shows how the signal links to the other elements.

Listing 5.6: Snippet of C# code

```
1 signals[0].Base = null;
2 signals[0].Head = points[0];
3
4 points[0].Face = signals[0];
5 points[0].Straight = tracks[0];
6 points[0].Turnout = tracks[1];
7
8 tracks[0].Left = points[0];
9 tracks[0].Right = null;
10
11 tracks[1].Left = points[0];
12 tracks[1].Right = null;
```

Similarly *Point* has three linking properties, *Face*, *Turnout* and *Straight*, which refer to the other elements as per the geographic position. In the listing 5.6 shows that

the point's *Face* is linked to *Signal*, whereas the other linkers, *Straight* and *Turnout*, linked to the respective tracks. Similar technique is applied to the track element as well.

The reason for implementing this technique is because initially the elements are unaware of which elements these are going to be linked to. Therefore, each child class is capable of linking to the desired number of linking elements. Once the geographical layout is known to program then the elements are linked to other elements correctly using their linking properties. Hence the classes in the Element inheritance hierarchy know their own name.

5.3 Conventional Approach

All railway stations comprise of the signalling elements as mentioned in Chapter 4 in Section 4.5. These elements are represented through different software objects within the program. This program requires the complete layout knowledge of the station for the algorithm to generate the control table. The geographical position of the signalling elements are loaded into the program through the technique explained in section 5.2.3. These objects of signalling elements have linking properties to link with other objects. The objects are linked together according to their geographical position of the layout. The linking properties of the software objects are drawn below in Figure 5.2.

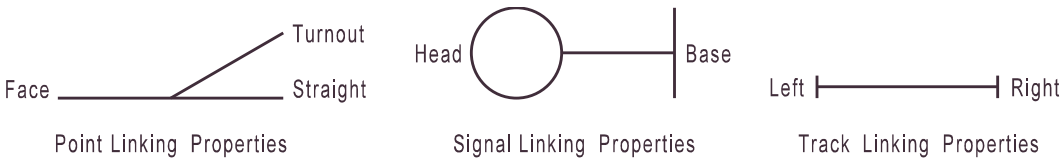


Figure 5.2: Elements Linking Properties

An example of a small section of the station is shown in the Figure 5.3a and linking of the elements of station is illustrated in the Figure 5.3b

(a) A small section of a station

(b) Linking the elements

Figure 5.3: Linking of the elements for a small section of the station

5.4 Compositional Approach

Unlike linking the elements of the station, compositional approach links the sub-sections of the station layout. Sub-sections are comprises of a set of elements in the station layout. Most often certain parts of the station layout looks similar to each other. Almost all the stations has some common sections within the layout that can be identified as the sub-section of that layout. These sections often re-appear exactly-as-it-is in other station layouts.

The most common and standard station layouts for signalling are identified in this research. These layouts are used to derive the common section of the layout to distinguish the sub-sections that are most commonly used in signalling. These sub-sections are linked together to represent the entire station layout. Hence different software objects are developed to present the sub-sections in the program. Four regularly used layout are considered to evaluate the sub-sections. These layouts are:

1. Single Turnout Station
2. Single Loop Station
3. Single Crossover Station
4. Double Crossover Station

The following Four sub-sections that are identified through the above mentioned four layouts:

1. Single Route
2. Loop Network
3. Single Crossover

4. Double Crossover

A detail descriptions of these sub-sections are provided in sections from 5.4.1 to 5.4.4. Sub-sections are made of elements and these elements are usually known with their general names within the South African railway. The dissertation make use of these general naming of the element to identify them.

5.4.1 Single Route

A station layout with only a single turnout is presented in the Figure 5.4. The layout shows the station starts with one mainline, thereafter making use of a point set, the mainline converts into double mainlines station. Point set provides an opportunity for the train to move to a second mainline. It is clearly recognised from the figure that the layout has a familiar element configuration at either end of the station. It is because normally all station will starts with *Intermediate Home Signal* going into the station and this signal followed by a *Home Signal* before reaching a point set. Normally there is an *Advance Starter signal* placed opposite of the intermediate home signal reading out of the station. The track in front of the advanced starter signal is known as *Approach Track* whereas the track after the advanced starter signal is known as *After Track*. The track going out of the station known as *Section Track*. The program will use this naming to identify the elements. The cut-line of the sub-section is shown in Figure 5.4.

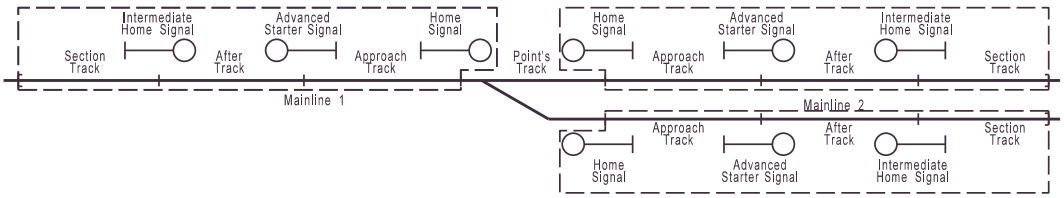


Figure 5.4: A typical example of a single turnout station with identifying names of all the elements

This research labelled this sub-section as *Single Route* because this sub-section entirely contains a single route from intermediate home signal to a home signal. The software linking properties of the sub-section are *Linked to Section Track* and *Linked to Home Signal Head*. The Figure 5.5 shows one of the linker is literally linking to the section track and other linker literally links to the head of home signal. The Figure 5.10 shows all the properties of the software object for the *Single Route* element.

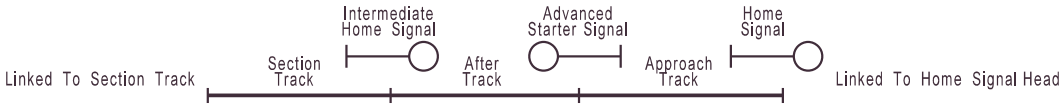


Figure 5.5: An example of a Single Route sub-section with the linking properties

Often there are distance signals placed just outside of the station to warn the driver that the driver is approaching the station. The distance signal normally doesn't entirely form part of the interlocking system. Therefore the distance signal is omitted in this research.

5.4.2 Loop Network

A station layout with only one loop is presented in Figure 5.6. The cut-line containing only the loop is clearly demarcated for the area of the loop network. Each loop network has two point sets at the either end of the loop. These points let the train admit into the loop and remit from the loop. There are normally four *Starter Signal* inside the loop. These signals are used for letting the trains exit the loop.

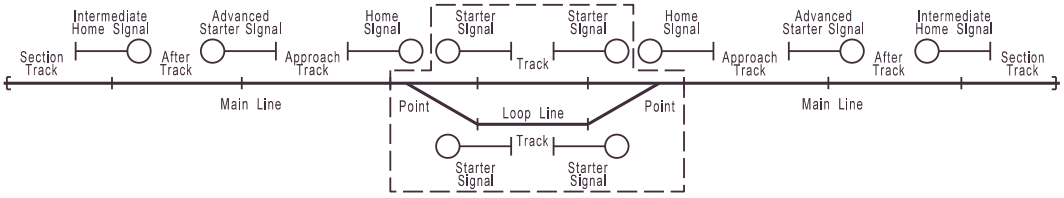


Figure 5.6: A typical example of a single loop station with identifying names of all the elements

For the software development the signals inside the loop need to named appropriately in the program and names of the signal cannot be the same. Therefore mainline starter signals are named as *Mainline Left Starter Signal* and *Mainline Right Starter Signal* whereas the loopline starter signals are named as *Loopline Left Starter Signal* and *Loopline Right Starter Signal*. The centre tracks within the loop is named as *Mainline Centre Track* and *Loopline Centre Track*. Often, there are more than one centre tracks for each lines. However, for simplicity the program is developed using only one centre track for each line. The left point set is named as *Left Point* and right point set is named as *Right Point*. The linking properties are *Linked to Left Point*

Face and *Linked to Right Point Face*. Figure 5.7 shows all the software properties of the *Loop Network* object with their subsequent names.

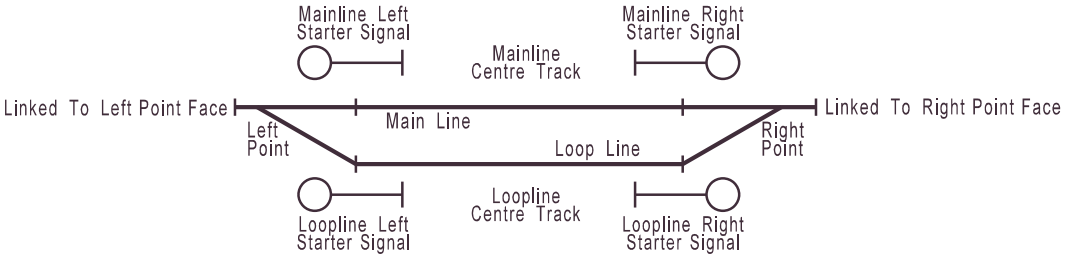


Figure 5.7: An example of a loop network sub-section with the linking properties

5.4.3 Single Crossover

A station layout with a single crossover is presented in Figure 5.8. Single crossover is comprised of two point sets. The purpose of the single crossover is to cross from one line to another line. The single crossover configuration in the Figure 5.8 shows train can only cross from the left of mainline 1 to right of mainline 2. This is not the only configuration that a single crossover can be built. The single crossover can also be configured such that train from left of mainline 2 cross to right of mainline 1. The cut-line of the single crossover sub-section area is demarcated in the Figure 5.8.

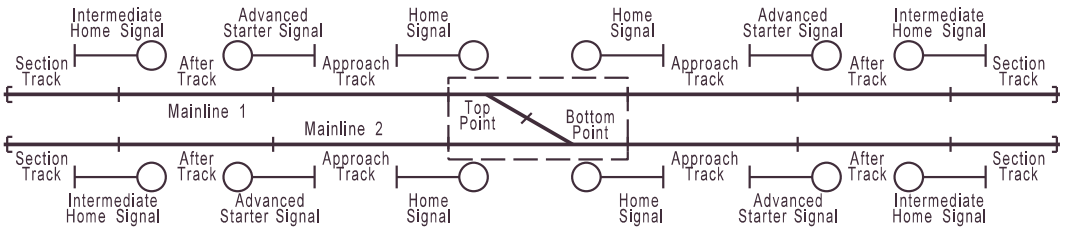
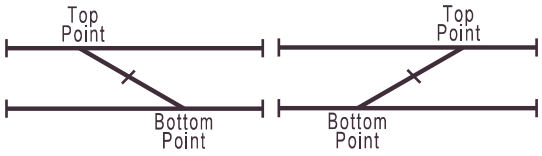


Figure 5.8: A typical example of a single crossover station with identifying names of all the elements

Figure 5.9 illustrates both configurations of the single crossover as explained above.



(a) Configuration 1 (b) Configuration 2

Figure 5.9: Both Configuration of the Single Crossover

The single crossover consists of two point sets. The mainline 1 point set is labelled as *Top Point* and the mainline 2 point set is labelled as *Bottom Point*. Subsequently the point's tracks name are *Top Point Track* and *Bottom Point Track*. The single crossover has four linking properties, they are, *Top Point Face*, *Top Point Straight*, *Bottom Point Face* and *Bottom Point Straight*. This single crossover software object satisfies for both configurations of the single crossover. Hence there is no need to create a separate software object for the program. Figure 5.10 shows all the software properties of the *Single Crossover* object with their subsequent names.

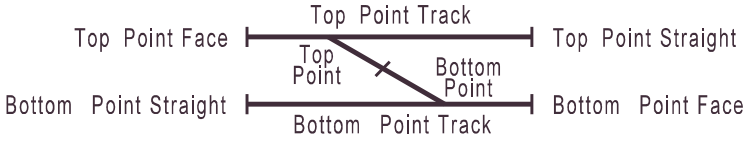


Figure 5.10: An example of a single crossover sub-section with the linking properties

5.4.4 Double Crossover

A station layout with a double crossover is presented in Figure 5.11. Double crossover contains four sets of point. Unlike single crossover the double crossover allows train movement from both mainline to cross to the other mainline. The double crossover configuration in the Figure 5.12 shows a method of building the crossover. But this is not the only configuration that a double crossover can be built. The cut-line of the double crossover sub-section area is demarcated in the Figure 5.11.

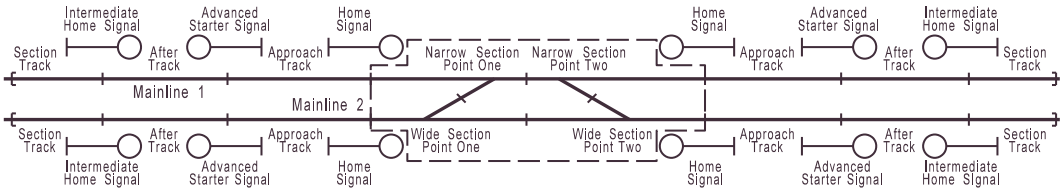
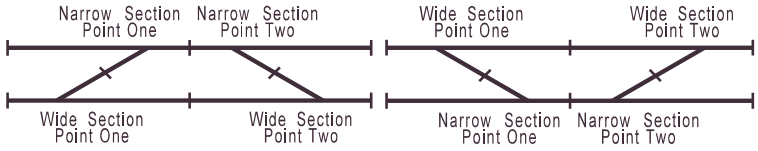


Figure 5.11: A typical example of a double crossover station with identifying names of all the elements

Both configurations of double crossover is presented in Figure 5.12.



(a) Configuration 1 (b) Configuration 2

Figure 5.12: Both Configuration of the Double Crossover

The double crossover consists of four point sets. These four points in double crossover always makes a trapezium shape between the mainlines where one of the mainline has the narrow side of the trapezium and the other line has the wide side of the trapezium. Observing this attribute of double crossover the elements were labelled in the program. The narrow section point sets are labelled as *Narrow Section Point One* and *Narrow Section Point Two*. The wide section point sets are labelled as *Wide Section Point One* and *Wide Section Point Two*. Subsequently the point's tracks name are *Narrow Section Point One Track*, *Narrow Section Point Two Track*, *Wide Section Point One Track* and *Wide Section Point Two Track*. The double crossover also has four linking properties, they are, *Narrow Section Point One Straight*, *Narrow Section Point Two Straight*, *Wide Section Point One Face* and *Wide Section Point Two Face*. This double crossover software object satisfies for both configurations of the double crossover. Hence there is no need to create a separate software object for the program. Figure 5.13 depicts the software object of the *Double Crossover*.

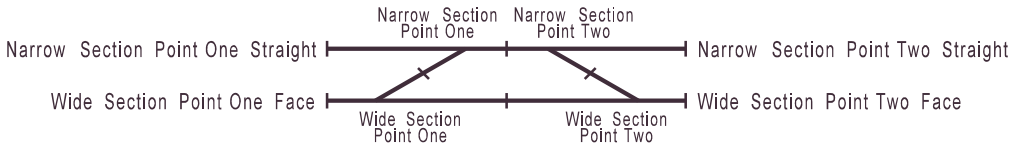


Figure 5.13: An example of a double crossover sub-section with the linking properties

Appendix A presents the above mentioned signalling layouts with actual signalling element numbers. The control table generating program auto-generates the text files with control table information that is also presented in appendix A.

5.5 Iteration Through Elements

One of the most essential parts of the program was to implement a method that allows the algorithm to navigate through the layout element to produce the items of the control table accurately. In order to explicate the method of navigation through the elements a simple layout will be discussed in this section. The following layout in Figure 5.14 is used to illustrate the program stepping through the elements.

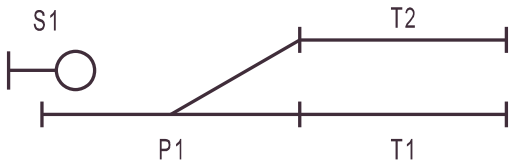


Figure 5.14: A portion of a layout

Once the station elements are defined within the program correctly, then the next step of the program is to link the elements to each other as per their exact geographical position on the layout. In Figure 5.14 shows a simple portion of the layout with one signal, one point and two tracks. As per the geographical layout the signal's, S1, linking property *Head* is connected to *Face* and linking property of the point, P1. Thereafter the linking properties of point, P1, *Turnout* is connected to the linking property *Left* of Track, T2, and subsequently *Straight* is connected to the linking property *Left* of Track, T1. It can be easily noted from the object linking diagram in 5.15 that the linking property of Signal, S1, *Base* and linking properties of Tracks, T1 & T2, *Right* is connected NULL value. It can be easily apprehend from the Figure 5.15 that the names of the elements are standard primitive *String* type and the linkers are *Element* type which is defined in this program.

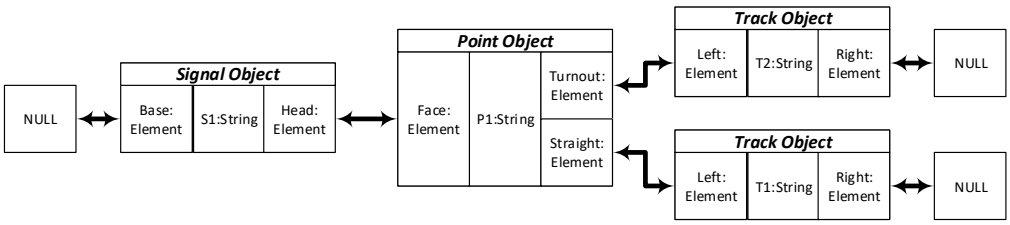


Figure 5.15: Linking the objects in software

In the first iteration of the program two temporary elements are used to traverse the layout. In Figure 5.16 shows for the route 1, *Current Element* is referring to S1 and *Next Element* is referring to P1.

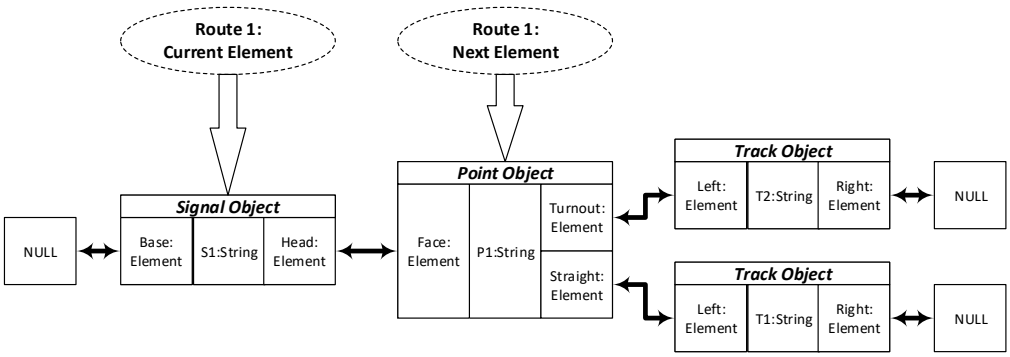


Figure 5.16: Traversing through the elements in first iteration

However in second iteration, the program realised that there are two possible routes from there onward. Figure 5.17 depicts two routes that has two set of temporary elements that are used to refer the elements of the different routes. The *Current Element* of route 1 is referring to P1 and *Next Element* of route 1 is referring to T2. Consequently the *Current Element* of route 2 is referring to P1 and *Next Element* of route 2 is referring to T1.

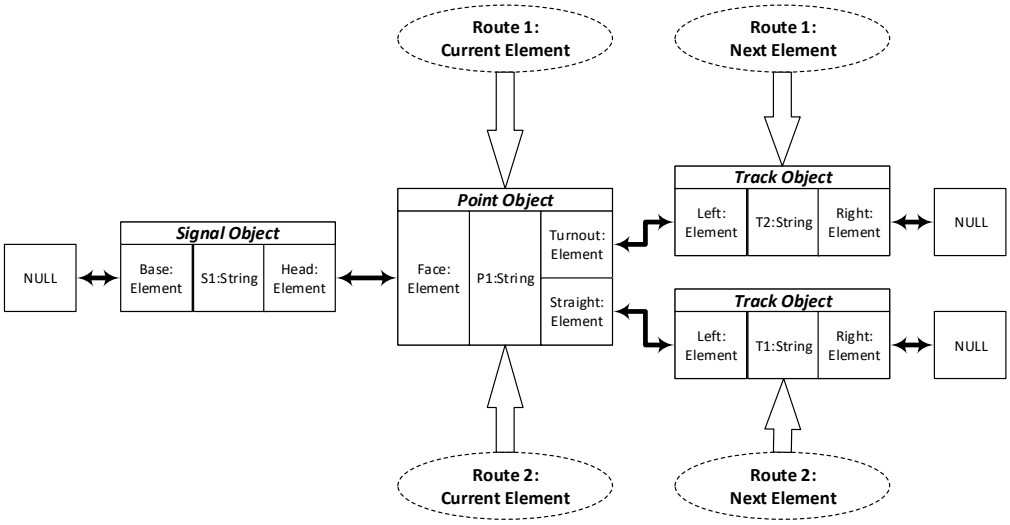


Figure 5.17: Traversing through the elements in second iteration

Chapter 6

Theoretical and Experimental Analysis

This chapter provides an analysis of the algorithms of auto-generating control table programs. The analysis was performed in data structure point of view. The kernel activities of the algorithms are explicated by making use of activity diagrams. The computational efficiency of the algorithms were investigated that assisted in gaining a broad understanding about the pros and cons of both algorithms, namely conventional and compositional approaches. The theoretical analysis enabled the development of general formulae which describe the performance of algorithms that assisted in predicting the experimental results. Finally the experimental results were documented in this chapter.

6.1 Analysis Overview

In data structure point of view the efficiency of an algorithm depends on the execution of the output data of the program for a arbitrary size of input. Should the algorithm of a task requires to execute a significantly large number of computing instructions than another algorithm of a same task then the first algorithm is relatively less efficient than the second algorithm. An improved algorithm shall be able to execute the same task requiring a much less number of computing instructions. In order to measure the success of an asymptotic analysis being considered. Big-O notation is an elegant method to conduct an asymtotic analysis. The Section 6.2 elaborates on the Big-O analysis of the generated programs.

Activity diagrams are one of the best approach to explicate the program at length.

The Section 6.3 describes the algorithms of the programs step-by-step. Firstly the main component of the programs are identified and thereafter the activities of the component are identified. Accurately identifying the activities enabled to estimate the operation for Big-O analysis. The main components of the software program are:

- Linking the Station Elements
- Generating the Route Information
- Generating the Flank Protection Information
- Generating the Aspect Information

The above proposed method of theoretical analysis aided to predict the expected results of program and this was supported by evaluating the general formulae. It is however essential to prove the accuracy of the formulae by conducting a experimental analysis. The actual number of computing operations were counted by the program that evidently proved the correctness of the formulae.

The analysis of the algorithms were conducted using two methods:

- Theoretical Analysis of Expected Result
- Experimental Analysis through Simulated Result

6.2 Big(O) Notation Analysis

The purpose of the data structure is to organise, modify and access data in a systematic way [50]. The efficiency of the algorithm depends on the effectiveness of the data structure, i.e. how successful the algorithm is in terms of organising the data effectively [51]. In data structure point of view the success of an algorithm is determined by the ability of the program to deal with large input data.

A definition of an algorithm of a software program can be stated as a finite sequence of instructions that can be executed with a finite amount of effort in a finite length of time [51]. The objective of an algorithm is follow a step-by-step procedure for accomplishing certain task in a finite amount of time [50]. Software programs are written using formal programming language which contains complex programming details that's taken care of by the computer while executing the program [51].

Analysis of program efficiency incorporates the usage of the computation resources,

they are CPU running time, use of memory storage, energy usage, data communication and etc. [52]. However the precise calculation of these resources provide the performance of the program within a particular environment, such as hardware environment and software environment [50]. In other words should the exact same program be executed on a different hardware or software then the performance of the program will be different.

Unlike programs, the algorithms are much simpler and hence it is possible to conduct an analysis without using any computer. Therefore analysis will only consider the design of the algorithms rather than analysing the entire programming details. Analysing the algorithm efficiency requires to calculate the complexity of the algorithm.

Since algorithm of a program comprises of a sequence of instructions, hence execution running time of each instruction will collectively represent the execution time of program. However it can be postulated that each instruction of the program requires a single unit of time whether the program execute an arithmetic operation, or a comparison, or accessing memory operation, etc. It is well understood in the domain of data structure that all operations cannot take same amount of time. Hence it is acceptable in data structure to calculate the number of operations for the calculation of the algorithm efficiency.

The complexity of the algorithm refers to the function that determines the number of operations executed by the computer for an input of arbitrary size [53]. Usually in theoretical analysis, the algorithm's complexity estimated in the asymptotic sense whereby the complexity function is estimated for arbitrarily large input [53]. Asymptotic analysis provides more insight on the algorithm's response for large inputs. An asymptotic analysis will be sufficient for this research work. Asymptotic algorithm analysis will be achieved through counting the operation executed and recording the result as a simple function using Big-O, Big-Omega, or Big-Theta notation [53]. In this research work calculations were arranged in terms of Big-O notation that required counting of the primitive operations of the algorithm. Primitive Operations of an algorithm can be given by the following:

1. Arithmetic calculation
2. Comparison
3. Variable declaration
4. Assignment statement
5. Indexing into an array

6. Invoking a method
7. Returning from a method

The snippet of code is shown in listing 6.1. The code is used to explain how the primitive operations are identified for this work to determine the number of operation executed by the algorithm.

Listing 6.1: Snippet of C# code

```
1  public void Starting_with_Route_Generation(Signal s)
2  {
3      Route initial_route = new Route();
4      routes.Add(initial_route);
5      initial_route.start_signal = s.ElementName;
6      initial_route.departure_signal_aspects = s.Signal_Aspects;
7      initial_route.route_number = routes.Count;
8      initial_route.current_element = s;
9      initial_route.next_element = s.Head;
10     initial_route.route_elements.Add(s);
11     Next_Element_in_Route(initial_route.next_element, initial_route);
12 }
```

- In line 3, an object of the class is declared and initialised. Therefore this line operates two operations.
- In line 4 and 10, adds an element to the list. This would be counted as one operation.
- From line 5 to 9, executes the assignment operation. Each assignment operation is counted as one operation.
- Line 11, invokes a method which is taken as one operation as well. However there are two arguments passed in that method which will perform two assignment operations with the parameters of the method. Hence in this line of the program will execute three operations.
- So in the given snippet of the code has a total of twelve operations.

The program uses extensive number of *foreach* loop. The snippet of the code in listing 6.2 explains how the number of operation counted within a *foreach* loop.

Listing 6.2: *foreach* loop

```
1 foreach (Signal s in sigs)
2 {
3     Starting_with_Route_Generation(s);
4 }
```

The code in the listing 6.2 shows, a *foreach* loop is used to invoke a method. The *foreach* loop is equivalent of *for* loop. The general form of the for loop is:

for(Initialisation, loop Continuation Condition, Increment)
    Statement

The example below shows the number of operation executed by the program for *for* loop:

for(int i = 0; i < n; i++)
{
    Body of the for loop
}

The primitive operations for only the *for* loop excluding the "Body of the loop" are:

- i is initialised once in the beginning of the *for* loop,
- the Loop Continuation Condition is executed $n + 1$ times where the condition were found to be true for n times and the condition was found to be false once when the *for* loop was exited
- incremental operation was executed for n times
- hence the total number of operation for only *for* loop is $2n+2$.

The program also make use of the *if* statement. The code in listing 6.3 is an example of the *if* statement used within the source code.

Listing 6.3: *if* statement

```
1 if (e.Base == r.current_element && r.next_element != null)
2 {
3     r.current_element = null;
4     r.next_element = null;
5     r.end_signal = e.ElementName;
6     r.destination_signal_aspects = e.Signal_Aspects;
7 }
```

- In line 1 of the listing 6.3 shows the *if* statement is used in the code. This statement has two cases with an AND operator. If the first case isn't true then the second case is not evaluated in C# . However, for this exercise this above *if* statement is counted as two operation because this is a high level Big-O

analysis which is already highly underestimated. There are a large number of instructions carried out by the commuter in machine level.

- From line 3 to 6, program executes four assignment operations.
- So in total this *if* loop executes six operations.

6.3 Activity Diagram

The activity diagrams are essential for a theoretical analysis. Mathematical equations can be derived through the activity diagram. These mathematical equations allow to predict the efficiency of the algorithms. The overview of the program with the main activities is given in the flow diagram of Figure 6.1.

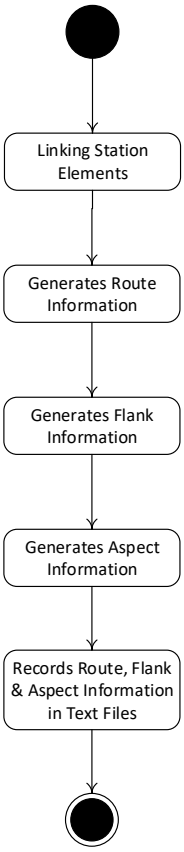


Figure 6.1: The flow diagram of main activities of the program

6.3.1 Linking Station Elements

The following diagram shows the activities carried out by the program to link the elements of the station

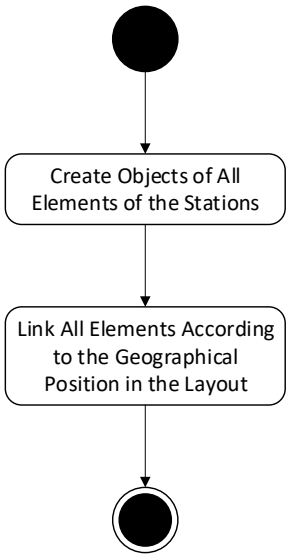


Figure 6.2: The activities of linking the elements of the station

It can be seen from the diagram above, there are two main activities for linking the elements of the station. These activities are, namely, creating the objects of the element and linking them in accordance with their geographical position in the layout.

Suppose C_{LE} represents the cost of the algorithm for linking the elements of the station. In other word the variable, C_{LE} is used to describe the number of operations required for the program to link the elements of the station.

Appendix A Figure A.1 shows the signalling diagram of a single turnout station. There are 19 signalling elements in the single turnout station layout. Therefore 19 objects of the elements need to be created in the program, thereafter these elements need to be linked by the program in a conventional approach.

Appendix A Figure A.2 shows the single turnout station with the compositional sub-sections cut-line. In compositional approach, the exact same layout is comprised

of only 4 sub-sections. Therefore the program only needs to create and link 4 objects of the sub-sections.

Now defining the variable C_{LE} for conventional approach as C_{LE-CON} and compositional approach as C_{LE-COM}, the equation (6.1) can be stated that:

$$C_{LE-CON} > C_{LE-COM} \tag{6.1}$$

6.3.2 Generate Route Information

A simplified flow diagram of generating the route information is illustrated in Figure 6.3. The initial node here refers to the start of route generation and final node refers to end of route generation.

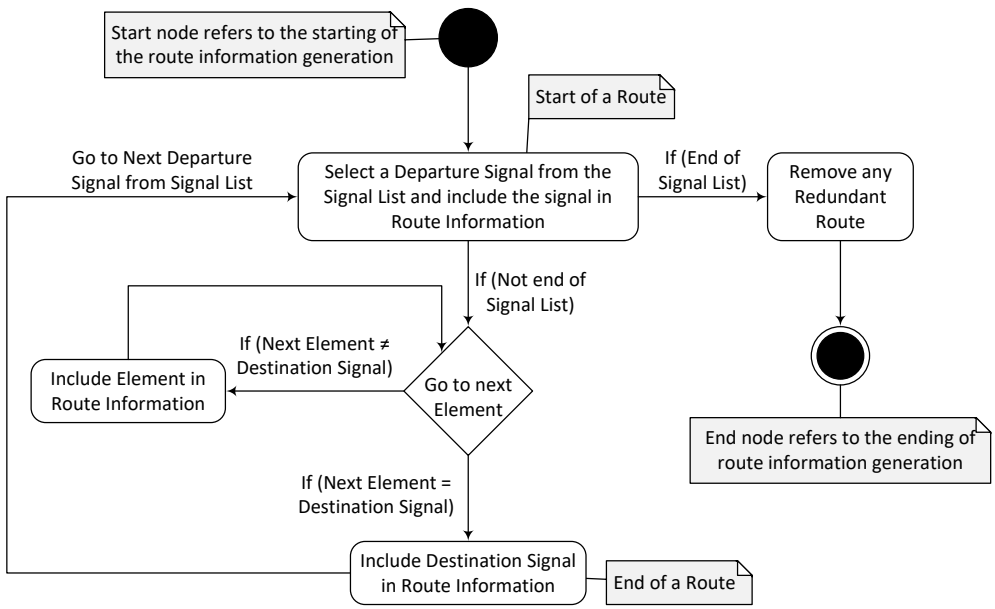


Figure 6.3: The simplified flow diagram of route generation

The flow diagram shows how each route is generated in Figure 6.3 where algorithm starts from the departure signal and ends with destination signal. The main components of the flow diagram are:

- Evaluating the route information from the departure signal to destination signal.

The flow diagrams in Figure 6.4 and 6.5 explains further on the inner steps of the route information evaluation.

- Removal of redundant route. The flow diagram in Figure 6.6 explains further on the detail steps of the removal of the redundant route.

Suppose R represents a set of routes of a layout where r indicates an element of the set R. Hence it is noted as $r \in R$ which means r belongs to R.

If there are n number of routes in the layout then the set R could be described as:

$$R = \{1, 2, 3, ...n\}$$

It is also shown in the flow diagram that algorithm iterate through all the elements in between the departure signal to the destination signal to evaluate the information of the route.

Let E be the number of elements of the route that algorithm require to iterate through from the departure signal to reach the destination signal. In other words there are E number of elements in the route including the destination signal but excluding the departure signal. It is because the algorithm starts iteration from the departure signal.

Therefore, E is a set of Natural number whereby E_r indicates an element of the set. Hence the set of E can be noted as:

$$E = \{E_1, E_2, E_3.....E_n\}$$

The details of the inner loop of the Figure 6.3 is illustrated in Figure 6.4 which shows how the linking properties of the element is used to navigate through the elements.

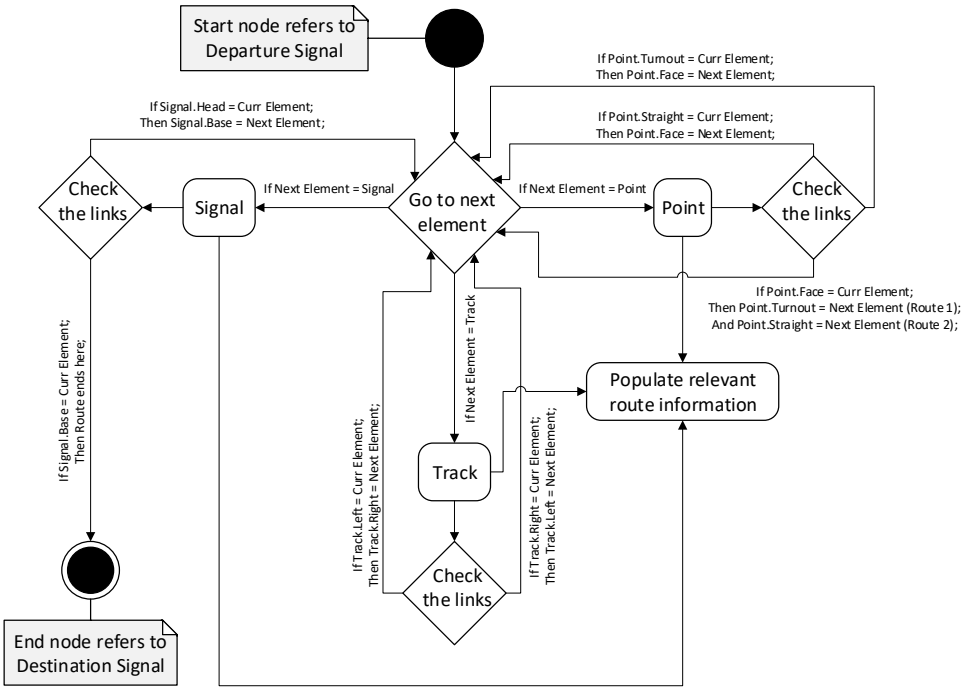


Figure 6.4: The flow diagram of the evaluating the elements of the route through traversing the layout

Chapter 5 Section 5.5 elaborates the method that program uses to iterate through the elements. The block diagrams in 5.16 and 5.17 depicts the technique that is used by the algorithm to move from one element to another. It is easier to comprehend the steps of the activities of the Figure 6.4 using the technique shown in 5.16. Two temporary elements are used, namely current element and next element, in this technique which are assigned to a new elements every time it iterates. This process requires some computing operations to be executed by the algorithm.

If T represents the number of operations executed by the algorithm to move through the elements which includes the operations such as, assigning temporary elements and invoking recursive functions and comparing through different types of elements. The number of operations need to be executed for the traversing process for only one route are:

$$E_r.T$$

Accumulatively the number of operations need to be executed for the traversing

process for n number of the routes are:

$$\sum_{r=1}^{n} E_r.T$$

Suppose the algorithm did not require all these operations of searching through the elements. In that case, the only operations that needed to be executed are to populate the route information in the correct routes which means assigning the element properties into route objects. This is shown in the figure 6.4 as an activity stating "populate relevant route information". This is the minimum number of operations required to evaluate the route information for a layout irrespective of whichever approach is used. A constant value K is assigned to represent the minimum number of operations required to evaluate the route information.

The total cost of route information is presented by C_{RI} then equation (6.2) can be derived through the activity diagram.

$$C_{RI} = \sum_{r=1}^{n} E_r.T + K \tag{6.2}$$

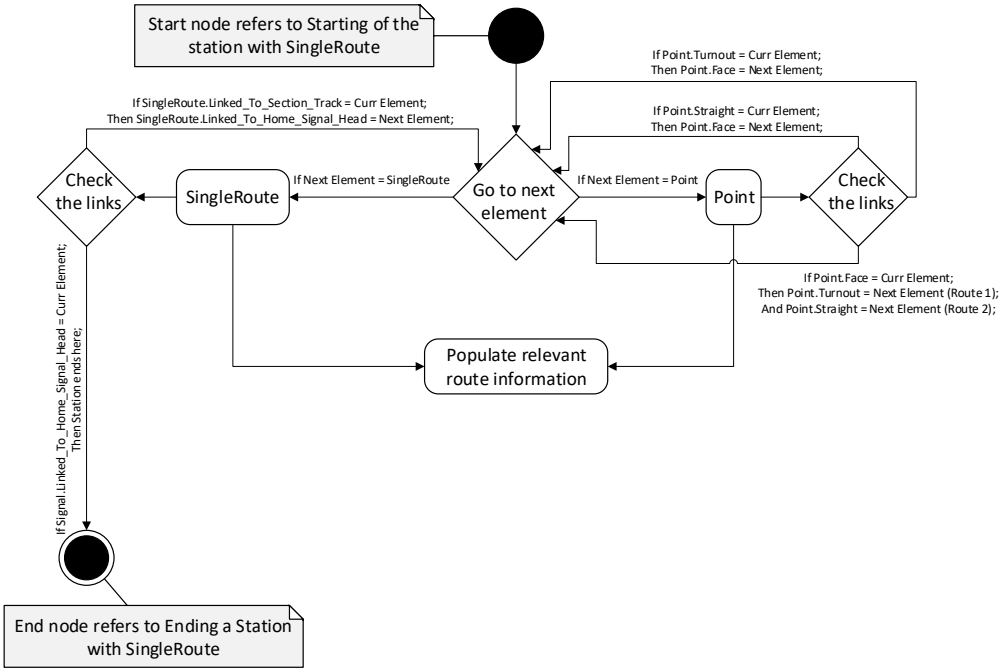


Figure 6.5: The flow diagram of the evaluating the elements of the route for Single Turnout Station

In order to compare the output of the equation given in (6.2) for conventional and compositional approach, the followings need to be considered. The variable with CON subscript refers to the variable in conventional approach and the variable with COM subscript refers to the variable in compositional approach.

- The variable K is constant for a layout irrespective of whichever method is used to evaluate the route information because K represent the least number of operations required to evaluate the route information. This means,

$$K_{CON} = K_{COM}$$

- The value of variable E_r in compositional approach must be less than the conventional approach for a exact same layout since the compositional approach reduces the number of elements. Hence the following statement must be true.

$$E_{r-CON} > E_{r-COM}$$

- In 6.5 shows there are only two types of software objects used in the single turnout station. Hence the program only need to compare within two types

of objects to determine which element is the next element. This suggest the number of operations require for this process will be less than the conventional approach because conventional approach has three types of objects. In other words, the value of T depends on the layout and number of types of element exist in that layout. However, there are other operations, such as, assigning the temp elements and invoking the recursive functions to evaluate the elements within the route which has to occur irrespective of the approach is used. Therefore for this analysis, it can be concluded that

$$T_{CON} \approx T_{COM}$$

From the analysis above it can concluded that cost of algorithm for conventional approach shall be more that compositional approach. Hence,

$$C_{RI-CON} > C_{RI-COM} \tag{6.3}$$

The removal of redundant route is shown in the flow diagram of 6.6.

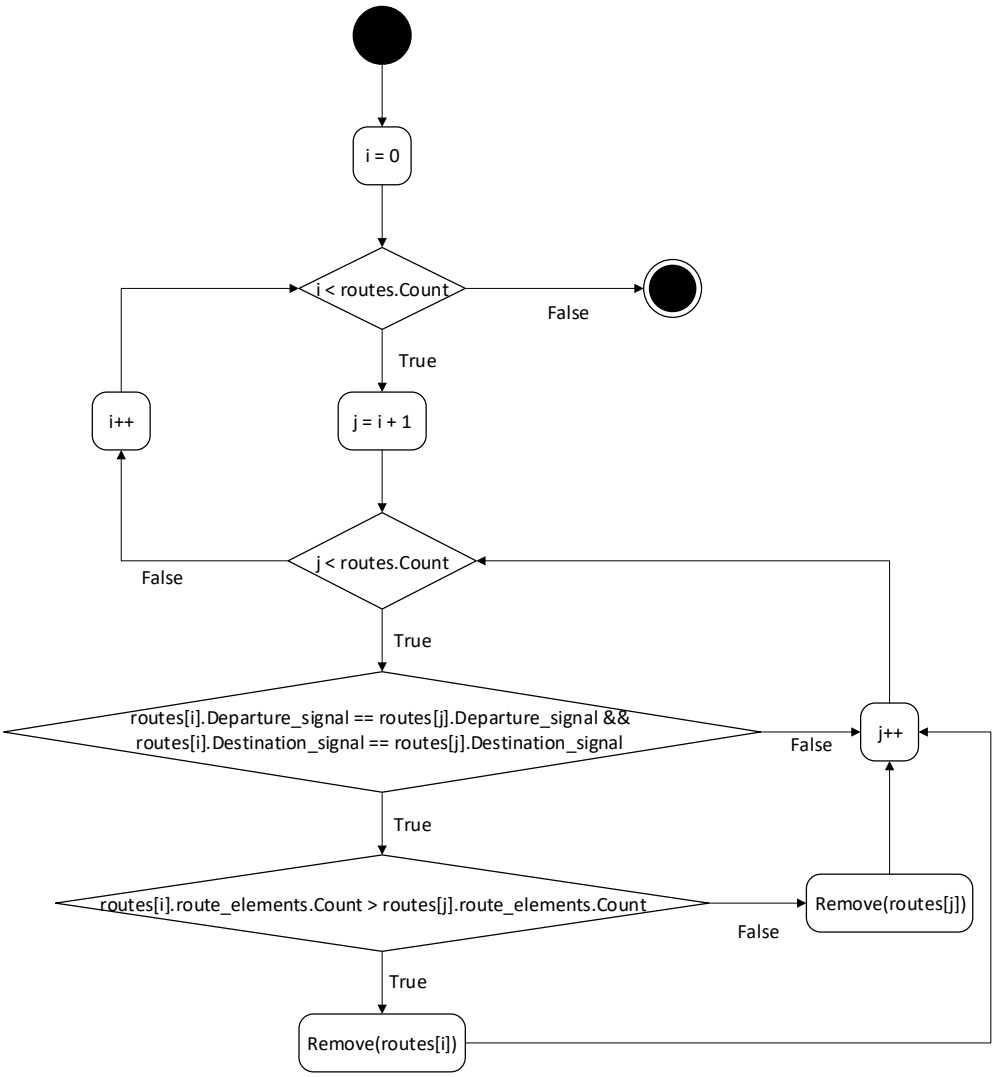


Figure 6.6: The flow diagram of removing the redundant route

The need for the removal of redundant route raised for the double crossover station layout. Chapter 4 Section 4.6, the signalling rule 11 states there are two possible routes for the same destination where one of them is straight route and the other one goes around the double crossover. The other route is not safe, hence it needs to be removed from the route list.

In the compositional approach, the program has knowledge of the sub-section of double crossover prior to evaluating the routes. Hence the program in compositional approach is able to omit this unsafe route prior to generating the routes.

Nevertheless in conventional approach the program does not have any knowledge of the double crossover in the layout. Therefore the program will generate the extra redundant unsafe route in the beginning and thereafter use this additional algorithm to remove this redundant route. In other words this additional algorithm is not required for compositional approach.

In Figure 6.6 shows there are two for loops in this portion of the algorithm. The activity diagram shows some of the variable used by the program. These variables are explained here:

- *routes* is a list where all the route object is stored.
- $routes.Count$ provides the number of routes in the layout.
- $routes[i]$ provides the route in the i^{th} position in the and similarly $routes[j]$ provides the route in the j^{th} of the *routes* list.
- $routes[i].Departure\_Signal$ provides departure signal of the i^{th} route and $routes[i].Destination\_signal$ provides the destination signal of the i^{th} route. Same applies for the j^{th} route as well.
- $route\_elements$ refers to the array which contains all the elements of the route
- $route\_elements.Count$ provides the number of elements in the route
- $Remove(routes[i])$ invokes the method $Remove()$ which removes the i^{th} route from the list. Same applies for $Remove(routes[j])$ as well.

The decision blocks of the flow diagram of removal of redundant route in Figure 6.6 is explained below:

- first decision block shows a for loop of variable i starts the iteration at $(i = 0)$ and ends the iteration at $(i < routes.Count)$. In other word, if the layout has n routes then this loop will iterate n times.
- second decision block shows a for loop of variable j
- third decision block checks $route[i]$ departure signal is same as the $route[j]$ departure signal and also checks $route[i]$ destination signal is same as the $route[j]$ destination signal. If $route[i]$ and $route[j]$ has the same departure signal and destination signal which means the starting and ending point of both route is same.
- if there are two routes with same departure signal and same destination signal then forth decision block checks for the route with most elements. The route with most elements goes around the bend in the double crossover which is the redundant route and need to be removed.

To evaluate the number of operations needed for this portion of algorithm to remove the redundant route requires to identify the number of times the main body of this activity diagram iterates. The *for* loop with variable i iterates n times.

if $i = 0$, then j loop starts at 1 and iterates $(n-1)$ times;
if $i = 1$, then j loop starts at 2 and iterates $(n-2)$ times;
if $i = 2$, then j loop starts at 3 and iterates $(n-3)$ times;
if $i = (n-2)$, then j loop starts at $(n-1)$ and iterates 1 times;
if $i = (n-1)$, then j loop starts at n and iterates 0 times;

Hence the total number of times these loops will iterate can be given by the series shown in (6.4).

$$(n-1) + (n-2) + (n-3) + ... + 1 = \frac{n^2 - n}{2} \tag{6.4}$$

There are two comparison tasks undertaken by the third decision block and there are one comparison task and one removing of a route task carried out by the forth decision block. In other word cost of main body of this algorithm in best case $O(2)$ and worst case $O(4)$. However, there are other internal operations in the for loops, such as, initialisation, loop continuation conditions and increments. Therefore let the variable V represent the number operations for removal of the redundant routes for each iteration. Then the total number of operations is calculated as

$$\left(\frac{n^2 - n}{2}\right) V \tag{6.5}$$

Including the above term to equation 6.2, The total cost of evaluating the route information becomes:

$$C_{RI} = \left(\frac{n^2 - n}{2}\right) V + \sum_{r=1}^{n} E_r.T + K \tag{6.6}$$

It is clear from the equation (6.6) that as the number of route increases the cost of operations for evaluating route information will increase quadratically. Hence the equation has a quadratic growth.

The term in 6.5 makes the cost of the algorithm of conventional approach much

greater than compositional approach. Therefore C_{RI-CON} becomes much greater than C_{RI-COM}.

6.3.3 Generate Flank Protection Information

A simplified flow diagram of generating the flank information is illustrated in Figure 6.7.

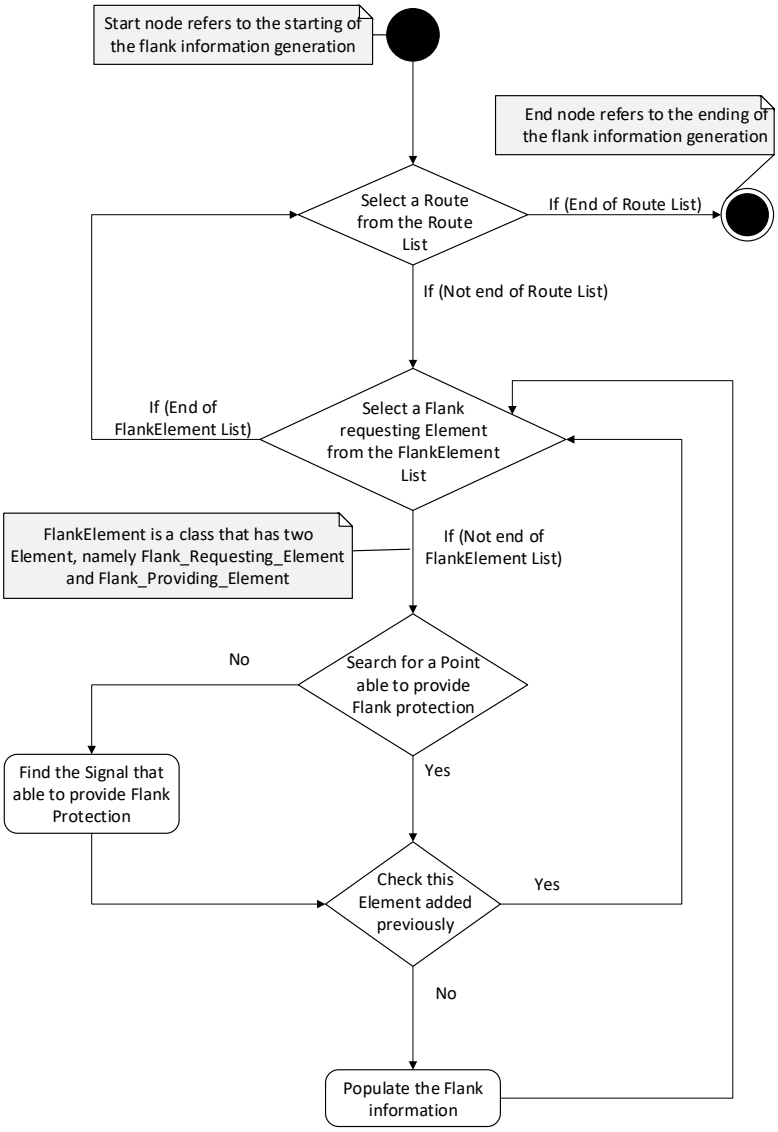
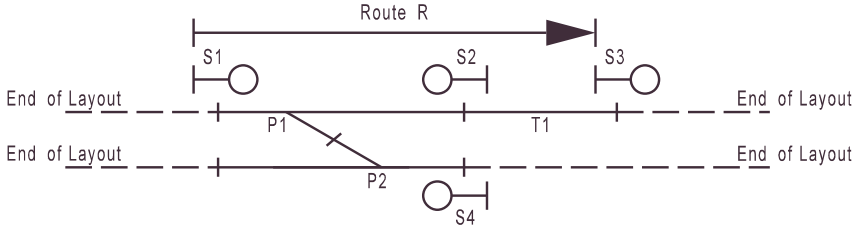


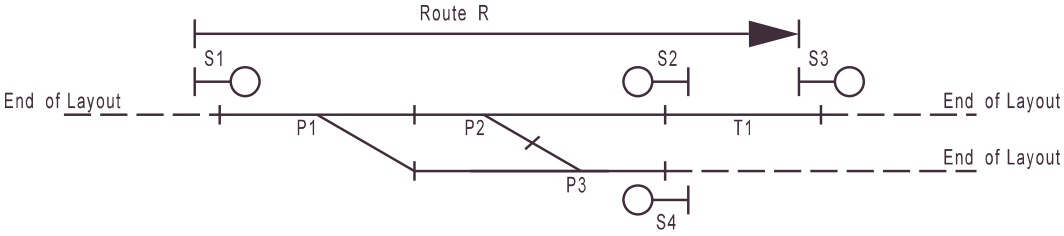
Figure 6.7: The simplified flow diagram of flank generation

The main components of the flank generation flow diagram are:

- Selecting the flank requesting elements. This is explained in details using the examples in Figure 6.8.
- Search for an element, for example, point or signal to provide flank protection. The detail activities are shown in the flow diagrams of 6.9 and 6.10.
- Checking this element added previously. This is to avoid adding same element as flank providing element more than once in the control table.



(a) The point set able to provide flank protection to the route



(b) The point set unable to provide flank protection to the route

Figure 6.8: An example of flank protection

The algorithm of evaluating flank protection information require objects of class *FlankElement*. The *FlankElement* class has two elements and they are:

- The element that requesting for flank protection which is Flank_Requesting_ELement
- The element that the flank protection requested to which is Flank_Providing_Element. Flank_Providing_Element does not necessarily provide flank protection. However, the searching of flank protection starts from this element.

A single route may require more than one flank protection, for example, in Figure 6.8a only point P1 requesting flank protection for route R. However in Figure 6.8b both points P1 and P2 requesting flank protection for route R. Therefore the Route

R in Figure 6.8a will have only one *FlankElement* object and the route R in 6.8b will have two *FlankElement* objects.

The layout in Figure 6.8 explains how point sets and signals provides flank protection. In Figure 6.8a shows point set P2 able to provide flank protection to Route R. In Figure 6.8b shows point set P3 is unable to provide flank protection to Route R since the straight and turnout path of P3 moving towards route R. Therefore the signal S4 needs to provide flank protection to Route R.

The algorithm prefers to obtain flank protection from the point set. Nevertheless, only and if only there is no point set that are able to provide flank protection then algorithm search flank protection from the signal as shown in the flow diagram 6.7. The algorithm begins searching for flank protection route-by-route. Therefore the algorithm of flank information will iterate n times for a layout with n^{th} route.

The flow diagram 6.9 shows the detail activities of the algorithm that ensures flank protection provided for a route.

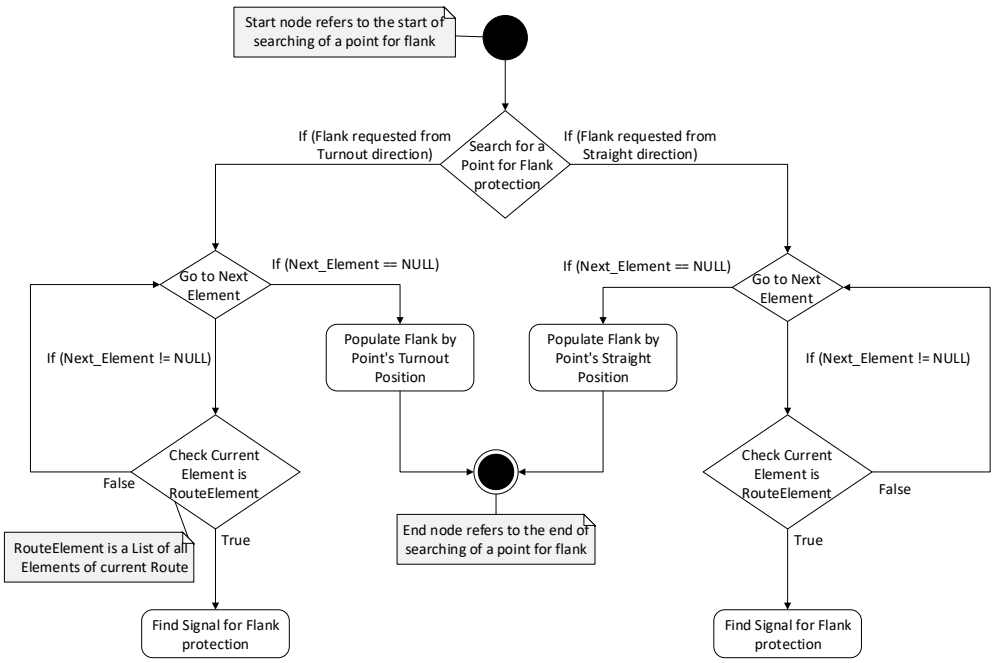


Figure 6.9: Providing flank protection

Each flank requesting element in the route need to search through other elements in the path for a possible flank providing element to ensure that train from this path

may not be able to collide with the train in the route. To ensure that the program needs to iterate through all the elements in the path of flank providing element till it reaches the end of the layout. This implies train in that path will not collide with the train in the given route. In the case where the possible flank providing element path coincide with route element then flank must be provided through signal.

Suppose F is a set of *FlankElements* objects per route. If there are m number flank protection required in the route then the set of F is noted as:

$$F = \{1, 2, ...m\}$$

For each flank requesting element, the algorithm needs to iterate through E_f number of elements. This implies the set of E_f is described as:

$$E_f = \{E_1, E_2, ...E_m\}$$

The decision block of "Check Current Element is RouteElement" in flow diagram of Figure 6.9 is further extended in flow diagram of Figure 6.10. This flow diagram shows detail activities regarding how each element is compared with route elements to ensure that the selected flank providing element doesn't belong to the route. In that case the selected point would not be able to provide flank protection.

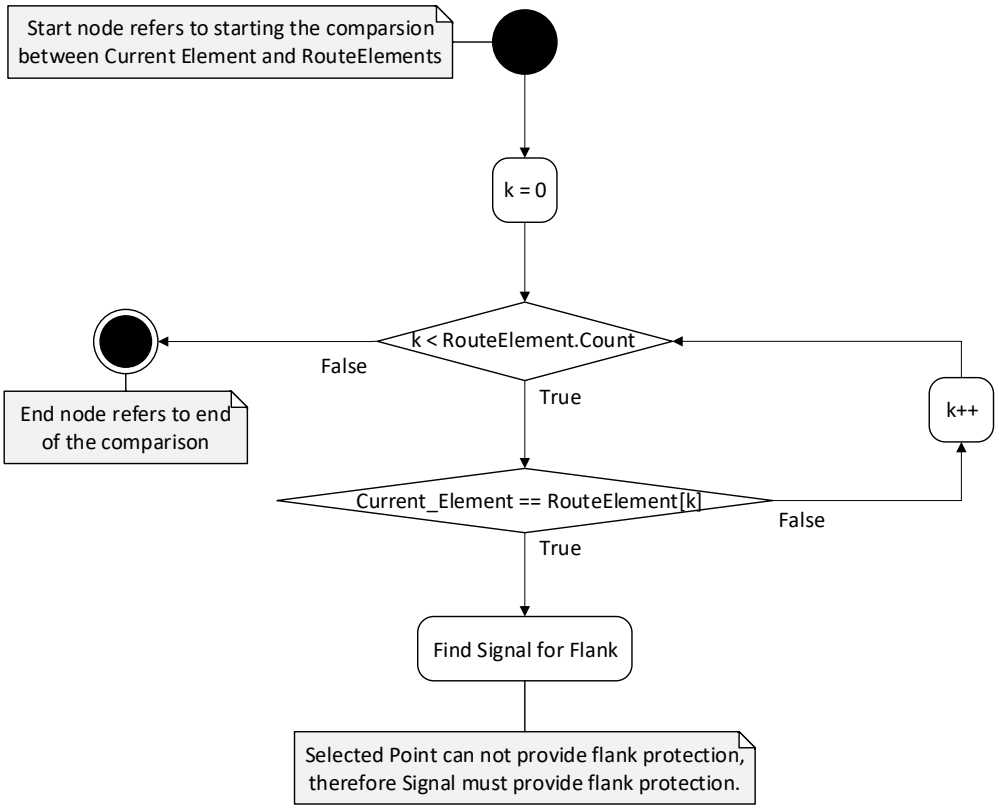


Figure 6.10: Ensuring the element doesn't belong to the route element

It stated in section 6.3.2 that E_r represents number of elements in the route excluding the departure signal. Therefore the total number of element in the route is $(E_r + 1)$. This implies that the loop in 6.10 will iterate $(E_r + 1)$ times.

The cost of the algorithm of the body of this loop will mostly be $O(1)$ since it just one comparison of the current element to the route element. However, when the current element is one of the route element then the point is unable to provide flank protection and a signal must provide flank protection. In that case, let s number of operations required for finding the signal to provide flank protection to the route. Hence the cost of algorithm of the body of the loop in worst case scenario will be $O(s + 1)$.

Let the variable S be the number of operations carried by the program each time it

iterates through this loop. The variable S can be defined as a function of two cases.

$$S = \begin{cases} O(1) & \text{if (Current\_Element == RouteElement[k]) is false} \\ O(s+1) & \text{if (Current\_Element == RouteElement[k]) is true} \end{cases} \tag{6.7}$$

Hence the total number of operations executed by this portion of the algorithm is:

$$(E_r + 1)S$$

This loop of comparing current element to the route elements will occur for all the elements in the path of probable flank providing element which is defined as E_f. Therefore the cost of this portion of the algorithm for evaluating a single flank of the route can be calculated as:

$$E_f(E_r + 1)S$$

The total cost of the algorithm to evaluate all the flank protections of the route shall be:

$$\sum_{f=1}^{m} E_f(E_r + 1)S$$

Let the variable P represent the number of operations for two activities of the flow diagram in Figure 6.7, they are:

- Check the element added previously
- Populate the flank information

In other words, if the algorithm did not require to search through the layout to evaluate the flank information even then at least P number of operations had to be executed.

The total cost of the algorithm for evaluating the flank protection information is represented by C_{FI} which is equal to:

$$C_{FI} = \sum_{r=1}^{n} \sum_{f=1}^{m} E_f(E_r + 1)S + P \tag{6.8}$$

The example is just one scenario where flank protection is required. There are many other scenarios or setup where flank protection is required from other elements. This example is used to explain the activity diagram. Other setup will require the modification of the implementation of the program to deal with the flanks. This example however enable us to compare conventional and compositional implementation thus providing more insight on the generation of flank protection information.

In order to compare the output of the equation given in (6.8) for conventional and compositional approach, the followings need to be considered. The variable with CON subscript refers to the variable in conventional approach and the variable with COM subscript refers to the variable in compositional approach.

- The variable P is constant for a layout irrespective of whichever method is used to evaluate the flank protection information because P represent the least number of operations required to evaluate the route information. This means,

$$P_{CON} = P_{COM}$$

- The value of variable E_r in compositional approach must be less than the conventional approach for a exact same layout since the compositional approach reduces the number of elements. Hence the following statement must be true.

$$E_{r-CON} > E_{r-COM}$$

- The value of variable E_f in compositional approach shall be less than the conventional approach for an exact same layout because the algorithm will need to iterate through less number of element to evaluate the flank protection. This implies,

$$E_{f-CON} > E_{f-COM}$$

- Essentially the variable S represents the number of operations required for checking the current element is not one of the route element. In both approaches the activity need to carried out and it should cost approximately the same. Hence,

$$S_{CON} \approx S_{COM}$$

From the analysis above it can concluded that cost of algorithm for conventional

approach shall be more that compositional approach. Hence,

$$C_{FI-CON} > C_{FI-COM} \tag{6.9}$$

6.3.4 Generate Aspect Information

The Figure 6.11 depicts the activity flow diagram of aspect generation.

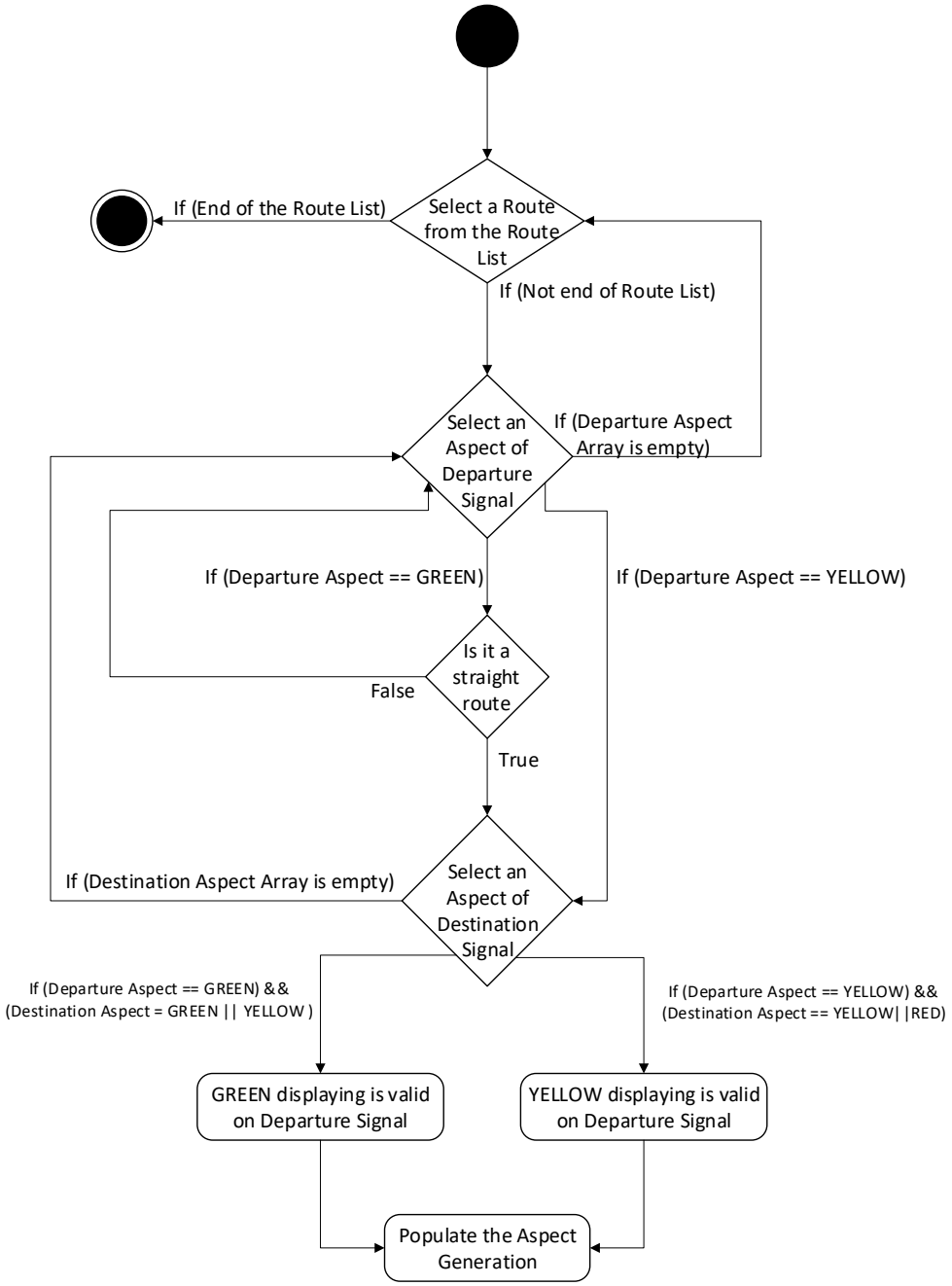


Figure 6.11: The flow diagram of aspect generation

The algorithm of for evaluating the aspect information doesn't depend on the number of elements of the route. The main activities of the algorithm are:

- Checking for what are the aspects available in departure signal and destination signal.

- Verifying if the route is straight route.

The departure aspect is driven through the destination aspect. In other words, the aspect of the current signal is driven through the aspect of the next signal. This is to inform the driver how to handle the train from current signal to next signal. The Table 6.1 shows the valid aspect displaying conditions.

Table 6.1: Valid aspect displaying conditions

Departure Signal Aspect	Destination Signal Aspect	Explanation
		A straight route and overlap available ahead of the destination signal. Therefore it is safe to display Green aspect on the departure signal. The driver may proceed at line speed.
		At least a straight route is available beyond the destination signal and there are no turnouts within this route which stretches from the departure signal to destination signal. The driver may proceed at line speed from the departure signal to destination signal.
		Case 1: There are turnouts between the departure signal and destination signal. Therefore driver must observe a slow speed while traversing over the turnout point set. Case 2: There are turnouts at the overlap ahead of the destination signal. Therefore Yellow aspect warns the driver to bring the train to a slow speed.
		Case 1: It is not a straight route ahead and there are turnouts within in this route. Therefore driver must travel at a slow speed from the departure signal to the destination signal. However, there is a straight route and overlap available beyond the destination signal. Case 2: If the Green aspect of the departure signal fails then the Yellow aspect can be displayed for the driver to move forward to the next destination signal on a straight route. This is to work on a degraded mode.
		The driver must obverse a slow speed so that the driver is able to stop short of the next signal

The algorithm uses the conditions of the Table 6.1 to correctly evaluate the valid

aspect for the departure signal for it's respective route. The following need to be considered to calculate the cost of the algorithm for generating the aspect information:

- Suppose there are U number of operations needed to evaluate the aspect displaying information.
- There are n number of routes in the layout.
- Let the variable C_{AI} be the total cost of algorithm for evaluation of aspect information.

The total cost of the algorithm for evaluation of aspect information is described using the function given in (6.10)

$$C_{AI} = nU \tag{6.10}$$

It is transparent from the analysis of evaluation of aspect information that the algorithm doesn't depend on the number of element used within the route. Therefore in both approaches, conventional and compositional, the cost of the algorithm will be exactly same. The variable with CON subscript refers to the variable in conventional approach and the variable with COM subscript refers to the variable in compositional approach.

$$C_{AI-CON} = C_{AI-COM} \tag{6.11}$$

6.3.5 Record the Test Files

The control table information are divided into route information, flank information and aspect information. These are the core information of the control table. Algorithm uses three separate text files to record these information. Appendix A presents all the program generated text files.

Recording of information doesn't form part of the core software of evaluating the critical information. This is only to write the information in a structured manner in test files. Therefore recording portion of program will not form part of the total cost of the algorithm of the control table generation. The core parts of the algorithm are, linking the station elements, generating the route, flank and aspect information.

The total cost of the control table generation algorithm is given by the equation (6.12), where C_{CTG} refers to the control table generation algorithm cost.

$$C_{CTG} = C_{LE} + C_{RI} + C_{FI} + C_{AI} \tag{6.12}$$

Considering the equations derived in (6.1), (6.3), (6.9) and (6.11), it can be concluded that cost of the conventional approach is much greater that the cost of the compositional approach to generate the control table. The variable with CON subscript refers to the variable in conventional approach and the variable with COM subscript refers to the variable in compositional approach.

$$C_{CTG-CON} > C_{CTG-COM} \tag{6.13}$$

6.4 Analysis of The Results

Appendix A presents all the diagrams of all the station layouts that were used in this research to generate the control tables. It also presents the automatically software generated control tables by the program using both approaches. The experiment was conducted on the seven layouts using conventional and compositional approaches. The cost of the algorithm was derived by counting the number of operations of the program. The number of operations were counted using few counting variables in the actual program. The variables were assigned for each counter respectively, for example, counting the linking operations, route operations, flank operations and aspect operations. The cost of algorithm for the control table generation for these layouts were recorded. From the table 6.2 to table 6.8 show the difference between the number of computation operations required by the program using conventional and compositional approaches for these layouts.

Table 6.2: The number of operations required to generate the control table of a single turnout station using different approaches

Layout: A Single Turnout Station		
Program Activities	Number of Operations Using Conventional Approach	Number of Operations Using Compositional Approach
Linking Station Elements	67	22
Generating Route Information	687	213
Generating Flank Information	506	226
Generating Aspect Information	466	466
Total	1726	927

Table 6.3: The number of operations required to generate the control table of a single loop station using different approaches

Layout: A Single Loop Station		
Program Activities	Number of Operations Using Conventional Approach	Number of Operations Using Compositional Approach
Linking Station Elements	72	19
Generating Route Information	864	238
Generating Flank Information	1614	42
Generating Aspect Information	710	710
Total	3260	1009

Table 6.4: The number of operations required to generate the control table of a single crossover station using different approaches

Layout: A Single Crossover Station		
Program Activities	Number of Operations Using Conventional Approach	Number of Operations Using Compositional Approach
Linking Station Elements	92	29
Generating Route Information	1010	310
Generating Flank Information	1776	232
Generating Aspect Information	762	762
Total	3640	1333

Table 6.5: The number of operations required to generate the control table of a double crossover station using different approaches

Layout: A Double Crossover Station		
Program Activities	Number of Operations Using Conventional Approach	Number of Operations Using Compositional Approach
Linking Station Elements	100	29
Generating Route Information	3734	438
Generating Flank Information	4578	430
Generating Aspect Information	862	862
Total	9274	1759

Table 6.6: The number of operations required to generate the control table of a complex station using different approaches

Layout: A Complex Station		
Program Activities	Number of Operations Using Conventional Approach	Number of Operations Using Compositional Approach
Linking Station Elements	232	102
Generating Route Information	37138	1920
Generating Flank Information	52916	3290
Generating Aspect Information	2606	2606
Total	92892	7918

Table 6.7: The number of operations required to generate the control table of a large station using different approaches

Layout: A Large Station		
Program Activities	Number of Operations Using Conventional Approach	Number of Operations Using Compositional Approach
Linking Station Elements	432	229
Generating Route Information	255844	4669
Generating Flank Information	290856	14815
Generating Aspect Information	5496	5496
Total	552628	25209

Table 6.8: The number of operations required to generate the control table of a line plan using different approaches

Layout: A Line Plan		
Program Activities	Number of Operations Using Conventional Approach	Number of Operations Using Compositional Approach
Linking Station Elements	1186	530
Generating Route Information	5121678	11112
Generating Flank Information	2940957	25375
Generating Aspect Information	14204	14204
Total	8078025	51221

The total number of elements of these seven layouts for conventional and compositional approaches are shown in following table.

Table 6.9: The number of elements required in different approaches

Station Layout	Number of Elements Using Conventional Approach	Number of Elements Using Compositional Approach	Number of Elements Reduced
Single Turnout	19	4	15
Single Loop	20	3	17
Single Crossover	26	5	21
Double Crossover	28	5	23
Complex	64	22	42
Large	118	53	65
Line Plan	330	118	212

It is clear from Table 6.2 to Table 6.8 that the number of the operations increase to generate the control table as the number of the elements increase. Percentages were calculated to determine the operations reduction efficiency for compositional approach as oppose to use the conventional approach. The table below shows the reduction percentage using the compositional approach

Table 6.10: The number of operations reduced in compositional approach in percentage

Station Layout	Total Number of Operations Using Conventional Approach	Total Number of Operations Using Compositional Approach	Number of Operations Reduced	Operations Reduced in Percentage
Single Turnout	1726	927	799	46.29%
Single Loop	3260	1009	2251	69.05%
Single Crossover	3640	1333	2307	63.38%
Double Crossover	9274	1759	7515	81.03%
Complex	92892	7918	84974	91.48%
Large	552628	25209	527419	95.44%
Line Plan	8078025	51221	8026804	99.37%

The graph below shows the number of operations reduced using compositional approach.

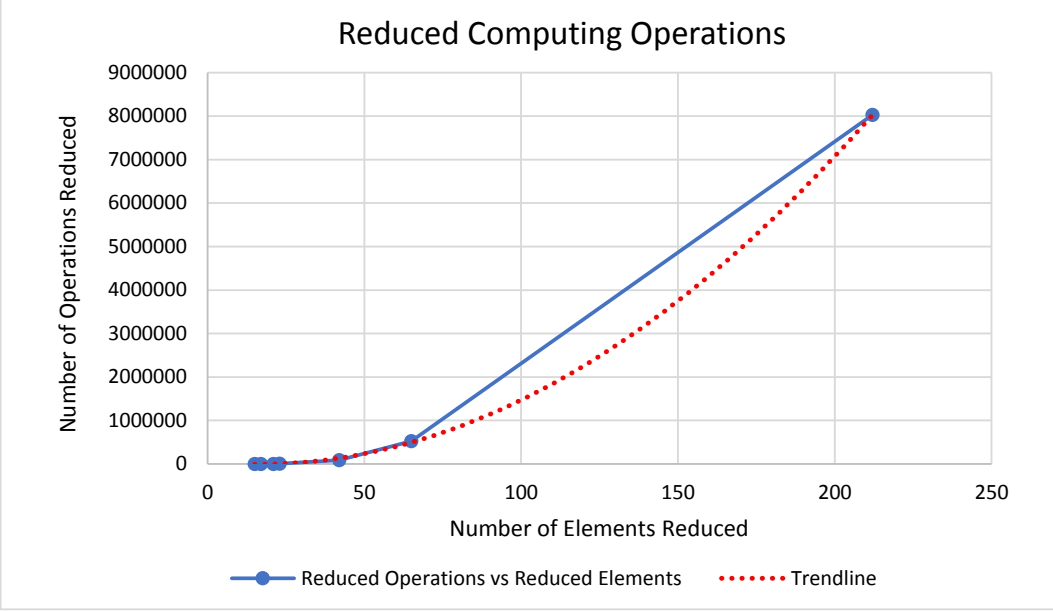


Figure 6.12: Comparison between the reduced number of elements vs the number of computing operations decreased

The blue line in Figure 6.12 shows the number of operations decreased in compositional approach compared to the conventional approach. The red line in the Figure 6.12 shows the quadratic trend line which best describes the behaviour of the program

in terms of reducing the computation operations. The equation 6.6 shows that the cost of the algorithm grows quadratically as the layout becomes larger. The graph in Figure 6.12 also has a quadratic growth which completely justifies the equation 6.6.

The graph below shows the reduction of the percentage of the computing operations

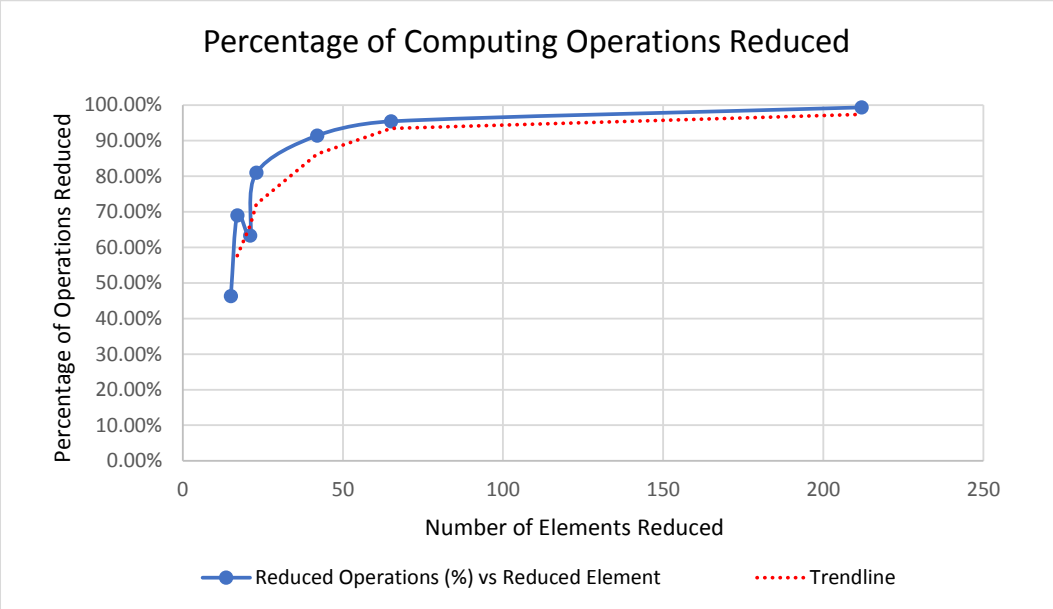


Figure 6.13: Comparison between the reduced number of elements vs the percentage of decreased computing operations

The blue line in the Figure 6.13 shows the percentage of operations decreased in compositional approach compared to the conventional approach. The red line in the Figure 6.13 shows the trend line which best describes the behaviour of the program in terms of reducing the computation operations in percentage because as the number of elements increasses the percentage of reduced operations will increase closer to 100%.

6.5 Statistical Analysis

A statistical analysis was conducted to gain a comprehensive understanding on the execution time of the programs and the time taken by the algorithm to generate the control tables. It was completely clear from the cost of the algorithm analysis that the compositional approach is more efficient than the conventional approach.

However, ideally this mathematical analysis of algorithm cost needs to correlate with the program's execution time for the better performance of the program.

A batch file code was developed to run the programs multiple times automatically. The control table generator program of both approaches was executed hundred times for each station layout. The batch file carried out a number of tasks in succession to automatically run the program hundred times. These tasks are listed below:

- Batch file initiates a for loop that runs hundred times,
- It records the start time,
- Then it runs the executable file of the control table generator program
- Then it runs the executable file of the control table generator program
- Thereafter it records the end time
- Finally it calculates the the duration of the program execution time
- Lastly a CSV file gets populated with the execution time

The shortcoming of the method was that batch files could only record the time in centiseconds. Therefore the centiseconds were multiplied with 10 to convert it to milliseconds. This means the execution time is accurate to 5 milliseconds.

The execution time can be divided into two parts. These are: the time taken by the system to include the necessary libraries and the actual algorithm time that evaluates the control table.

The algorithm time was recorded using the C# program in visual studio. Unlike the batch file, the visual studio IDE was able to output the timestamp with very high accuracy in milliseconds up to five decimal places. Hence the calculation of the algorithm time was highly accurate. As much as the total execution time is important in a holistic sense, the research's main focus is to obtain the algorithm time as accurately as possible.

Similar to the batch file, a timestamp variable recorded the start time of the algorithm in C# programming in visual studio IDE and another timestamp variable recorded the end time of the algorithm. Using these timestamps, the duration of the algorithm time was calculated which was written to a CSV file.

Appendix D provides the execution time and the algorithm time of the multiple runs of the executable files using both approaches for all the station layout. The average, maximum and minimum of the execution time and algorithm time was recorded for

each station layout using both approaches. The Table 6.11 to Table 6.17 shows the average, maximum and minimum execution time and the algorithm time of both approaches of all the layouts.

Table 6.11: The execution time and algorithm time taken to generate the control table of a single turnout station using different approaches

Layout: A Single Turnout Station				
	Conventional Method		Compositional Method	
	Execution Time (ms)	Algorithm Time (ms)	Execution Time (ms)	Algorithm Time (ms)
Average	76.00	16.46	75.90	16.28
Maximum	140.00	23.99	90.00	25.98
Minimum	60.00	14.99	60.00	14.98

Table 6.12: The execution time and algorithm time taken to generate the control table of a single loop station using different approaches

Layout: A Single Loop Station				
	Conventional Method		Compositional Method	
	Execution Time (ms)	Algorithm Time (ms)	Execution Time (ms)	Algorithm Time (ms)
Average	78.40	17.97	85.70	16.72
Maximum	130.00	46.87	170.00	31.25
Minimum	60.00	15.61	60.00	15.62

Table 6.13: The execution time and algorithm time taken to generate the control table of a single crossover station using different approaches

Layout: A Single Crossover Station				
	Conventional Method		Compositional Method	
	Execution Time (ms)	Algorithm Time (ms)	Execution Time (ms)	Algorithm Time (ms)
Average	78.50	18.28	73.90	16.72
Maximum	170.00	34.59	130.00	46.87
Minimum	30.00	15.62	60.00	15.60

Table 6.14: The execution time and algorithm time taken to generate the control table of a double crossover station using different approaches

Layout: A Double Crossover Station				
	Conventional Method		Compositional Method	
	Execution Time (ms)	Algorithm Time (ms)	Execution Time (ms)	Algorithm Time (ms)
Average	100.50	24.41	101.80	16.72
Maximum	300.00	75.02	280.00	152.48
Minimum	30.00	18.99	30.00	15.99

Table 6.15: The execution time and algorithm time taken to generate the control table of a complex station using different approaches

Layout: A Complex Station				
	Conventional Method		Compositional Method	
	Execution Time (ms)	Algorithm Time (ms)	Execution Time (ms)	Algorithm Time (ms)
Average	95.40	32.89	93.30	28.09
Maximum	200.00	93.75	380.00	33.98
Minimum	70.00	15.62	70.00	15.62

Table 6.16: The execution time and algorithm time taken to generate the control table of a large station using different approaches

Layout: A Large Station				
	Conventional Method		Compositional Method	
	Execution Time (ms)	Algorithm Time (ms)	Execution Time (ms)	Algorithm Time (ms)
Average	125.00	49.15	107.60	40.70
Maximum	280.00	125.00	240.00	63.28
Minimum	90.00	31.25	90.00	37.98

Table 6.17: The execution time and algorithm time taken to generate the control table of a line plan using different approaches

Layout: A Line Plan				
	Conventional Method		Compositional Method	
	Execution Time (ms)	Algorithm Time (ms)	Execution Time (ms)	Algorithm Time (ms)
Average	178.60	107.48	124.80	56.11
Maximum	360.00	127.92	230.00	69.96
Minimum	160.00	95.94	110.00	51.97

Once the execution time and the algorithm time was obtained using the batch file and visual studio, then it was straightforward to calculate the time taken by the system to include the libraries. The time taken for the inclusion of the system libraries should not vary to a great extent, because the program needed to include the same standard libraries irrespective of the size of the layout and the approach taken to generate the control table. Whereas the algorithm time is completely dependent on the size of the layouts.

The time taken by the system is denoted by T_{sys}, the execution time is denoted by T_{exe} and the algorithm time is denoted by T_{alg}. Therefore, T_{sys} can be given by:

$$T_{sys} = T_{exe} - T_{alg}$$

Table 6.18 shows the average time taken by the system to include the necessary libraries for the program execution. The minimum average time was found to be 59.54 ms and the maximum average time was found to be 80.69 ms. While program mostly took approximately 60 to 70 ms to include the system libraries for the program execution.

Table 6.18: The time taken by the system for various stations using different approaches

Station Layouts	Time Taken By The System (ms)	
	Conventional Method	Compositional Method
Single Turnout	59.54	59.62
Single Loop	60.43	68.98
Single Crossover	60.22	57.18
Double Crossover	76.09	80.69
Complex	62.51	65.21
Large	75.85	66.90
Line Plan	71.12	68.69

Table 6.19 shows the average algorithm time of both approaches for all the station layouts. The blue line in Figure 6.14 shows the average time taken by the control table generator program using the conventional approach. The orange line in Figure 6.14 shows the average time taken by the control table generator program using the compositional approach. It is absolutely clear from the graph in Figure 6.14 that the program with conventional approach took longer algorithm time than the compositional approach.

Table 6.19: The algorithm time for various stations using different approaches

Station Layouts	Algorithm Time (ms)	
	Conventional Method	Compositional Method
Single Turnout	16.46	16.28
Single Loop	17.97	16.72
Single Crossover	18.28	16.72
Double Crossover	24.41	20.91
Complex	32.89	28.09
Large	49.15	40.70
Line Plan	107.48	56.11

The graph in Figure 6.14 shows the average algorithm time of various station layouts.

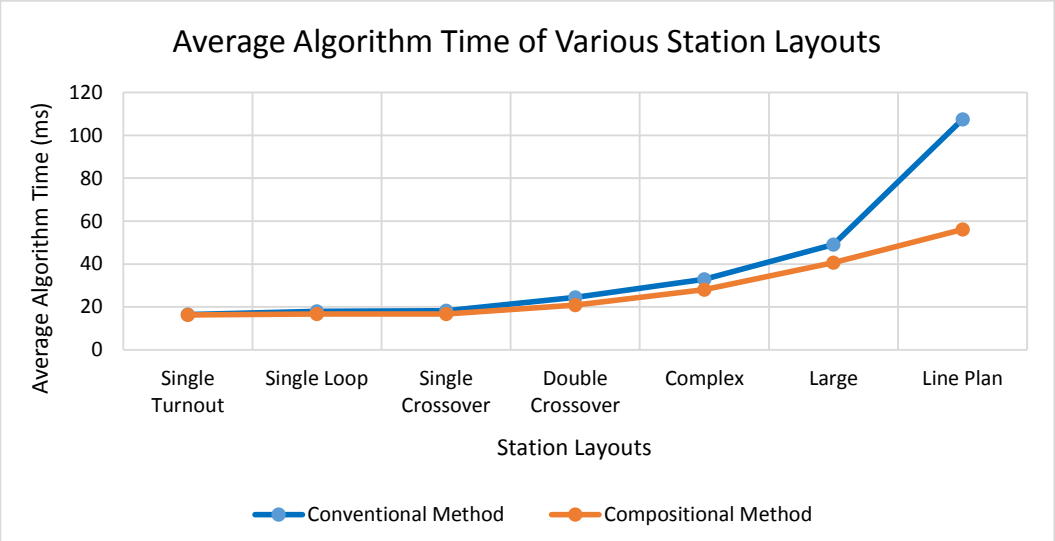


Figure 6.14: Comparison between the algorithm time of conventional method and compositional method for various stations

In order to determine the efficiency of the compositional approach, the reduction of the algorithm time of the compositional approach in comparison to the conventional approach was calculated. Thereafter the percentage of the reduction of the algorithm time was calculated to find the actual efficiency of the program in terms of the running time of the algorithm.

The reduced average algorithm time can be calculated by subtracting the average

algorithm time of compositional approach from the average algorithm time of conventional approach. Reduced average algorithm time is denoted by $T_{reduced-alg}$, the average algorithm time of convetional approach is denoted by $T_{alg-con}$ and the average algorithm time of compositional approach is denated by $T_{alg-com}$. Therefore the reduced average algorithm time is given by:

$$T_{reduced-alg} = T_{alg-con} - T_{alg-com}$$

The algorithm efficiency can be calculated using the following equation:

$$Algorithm\ Efficiency = \left(\frac{T_{reduced-alg}}{T_{alg-con}} \times 100 \right) \%$$

Table 6.20: The improved efficiency of algorithm time for various stations

Station Layouts	Reduced Algorithm Time (ms)	Algorithm Efficiency (%)
Single Turnout	0.18	1.09
Single Loop	1.25	6.96
Single Crossover	1.56	8.52
Double Crossover	3.50	14.32
Complex	4.80	14.60
Large	8.45	17.19
Line Plan	51.37	47.79

Table 6.20 shows the reduced algorithm time and algorithm efficiency for all the station layout used in this research. The graph in Figure 6.15 shows the algorithm efficiency has the trendline of parabolic shape indicating the quadratic growth rate of the program's efficiency. The efficiency of the program increases by reducing the algorithm time which correlates with the graph in Figure 6.12 that also has the trendline of a parabolic growth where number of operations were reduced for compositional approach. The efficiency percentage of algorithm time is shown in the graph 6.15.

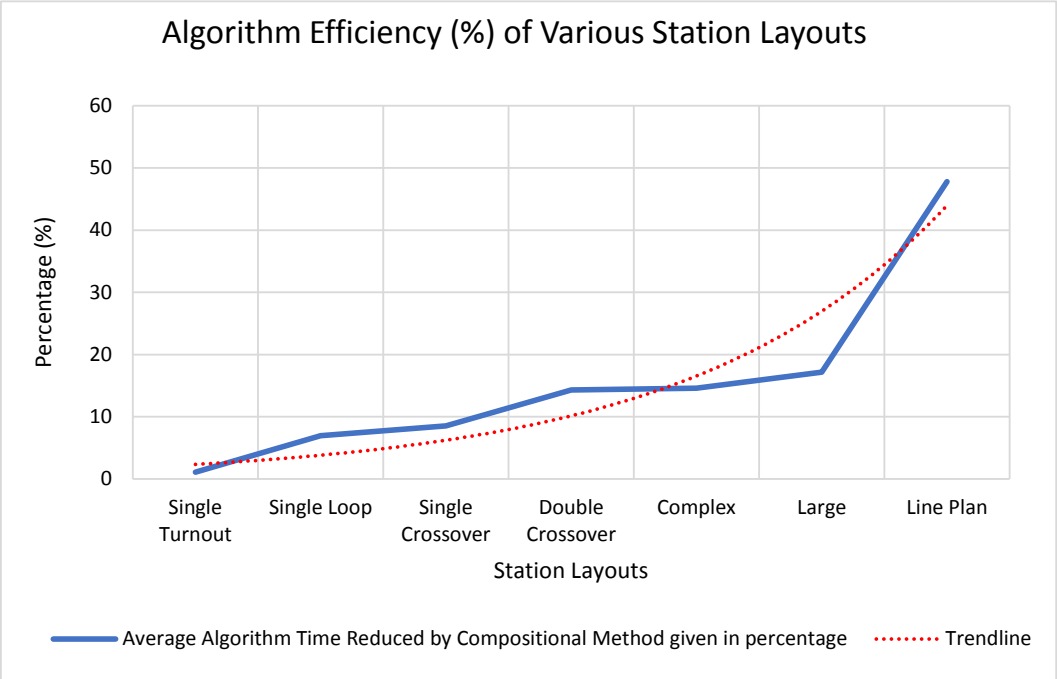


Figure 6.15: The percentage of algorithm efficiency

It can be concluded through the analysis that the decrease in number of operations will decrease the algorithm time making the compositional approach a efficient method of control table generation.

Chapter 7

Conclusion and Future Improvements

Auto-generation of Control table has been an intriguing research area for the signalling engineers for decades. Recently new methodologies are applied to generate the control table in more efficient ways. Compositional Approach has been most prevailing method for the signalling research groups throughout the world since researchers accomplished success in verifying the control table using this approach. However, there is not enough evidence of using the compositional approach to generate the control table. This initiated the concept of generating the control table using compositional approach.

This dissertation presented the approaches, namely conventional approach and compositional approach, that were applied to generate the interlocking control table. Detailed description of the processes of implementation of control tables for both approaches were provided. The efficiency of the algorithms were compared and presented. The analysis and result of the programs for both approaches were explained at length.

The efficiency of the algorithms were measured in terms of data structure perspective. In terms of determining the efficiency of an algorithm Big-O notation provides an elegant and useful practical option. It was possible to derive the general equations using Big-O notation that proved compositional approach is a better choice to generate the control table.

The research covers different areas of knowledge, such as, railway signalling, software engineering, comparative studies and statistics. It was realised while covering the diverse knowledge area for the research that some of the aspects were overlooked and required improvement. Often the best coding practice was not applied while

implementing the software program, such as, excessive use of *public* access modifiers, which could have been changed to *protected* or *private*, access modifiers.

Initially it was not envisaged that the Element class of the compositional approach would become excessively large. This created an opportunity to make the program more modular based with smaller classes. However, the research focused on comparative studies, where it is important to design the programs similar to each other. In other words, they use the same number of classes and similar types of classes to generate the control table. If one program uses a greater number of classes than the other program then that defeats the purpose of like-for-like comparative analysis. Moreover making the compositional approach more modular based may assist in further optimisation in the runtime performance which will be an unfair comparison to the program of conventional approach. Nevertheless, It was perceived that in terms of the software development point of view, it is imperative to apply good coding practice which can be an aspect of future improvement of the program.

This research shows great benefit was achieved through producing the control table compositionally in terms of algorithm efficiency. It can be deduced based on the empirical analysis that the reduction of the number of operations directly influences the reduction of the algorithm running time. Based on the results and the analysis it can be convincingly concluded that the outcome of the research is that the compositional approach is more efficient than the conventional approach.

Software programs were developed using a general purpose high level programming language for this research work. Several research groups worked on domain specific programming software to verify the control table. Should the research be conducted using a railway domain specific software to automatically generate the control table then intuitively compositional approach is more probable to perform better in performance-wise.

This investigation was carried out using some of the basic signalling rules and regulations. There are many scopes for extending these rules and regulations, for example, the overlap conditions may be included, multi-aspect signalling may be considered, etc. However, inclusion of these features will make the implementation of the program more complex.

It was observed that the greater the number of types of sub-section available to use in the program the better the computational efficiency for the algorithm. Hence more

types of sub-sections need to be identified from the standard patterns on the station layout. An investigation shall be undertaken to derive the control table using these new types of sub-sections for a range of more complex layouts. One of the drawback for this approach is that each sub-sections need to be implemented for the program which is time consuming.

The current programs were developed using the standard layouts. More often, the station layouts deviate from the standard layouts. For these special cases program need to be modified to accurately evaluate the control table. This makes the program inflexible to make modifications. The future research may concentrate on designing the program that more flexible to these modifications for compositional approach.

The main purpose of the investigation was to investigate on finding a efficient approach to auto-generate control table. The research work demonstrated that indeed the compositional approach is more efficient to use for the control table generation in terms of the number of instructions to be computed by the program and the actual time taken by the algorithm to evaluate the control tables.

References

[1] Imad Zara. *Methods, Notation and Tools for Modelling Command and Control Systems*. Version 1.0. POR CReO FESR 2007-2013, LINEA D'INTERVENTO 1.5a - 1.6, BANDO UNICO R&S ANNO 2012, Co-founded by Tuscany Region. Department of Information Engineering (DINFO), University of Florence, 2012.

[2] Imad Zara. *Report of a Comparative Analysis of the Interlocking Systems*. Version 1.0. POR CReO FESR 2007-2013, LINEA D'INTERVENTO 1.5a - 1.6, BANDO UNICO R&S ANNO 2012, Co-founded by Tuscany Region. Department of Information Engineering (DINFO), University of Florence, 2013.

[3] Colin White. *Interlocking Principles*. London Underground Ltd, 2010.

[4] Stephen Clark. *A History of Railway Signalling (from the Bobby to the Balise)*. 71 Fenechurch street, London EC3M 4BS: *Lloyd's Register Rail, UK.

[5] Ahmad Mirabadi and Mohammad B. Yazdi. *Automatic Generation and Verification of Railway Interlocking Control Tables using FSM and NUSMV*. Iran University of Science and Technology, School of Railway Engineering (S.R.E), Iran, Tehran, Narmak, 2009.

[6] David Tombs, Neil Robinson and George Nikandros. *Signalling control table generation and verification*. RTSA. Conference on Railway Engineering Wollongong, 2002.

[7] Alessandro Fantechi, Anne Elisabeth Haxthausen and Hugo Daniel dos Santos Macedo. "Compositional Verification of Interlocking Systems for Large Stations". In: *Software Engineering and Formal Methods, Springer, Lecture Notes in Computer Science* 10469 (2017), pp. 236–252. DOI: http://doi.org/10.1007/978-3-319-66197-1_15.

[8] Oytun Eris and ilhan Mutlu. "Design of Signal Control Structures Using Formal Methods for Railway Interlocking Systems". In: *11^{th} International Conference Control, Automation, Robotics and Vision, Singapore* (2010).

[9] Uğur Yıldırım, Mustafa S. Durmuş and Mehmet T. Söylemez. *Automatic Interlocking Table Generation for Railway Stations using Symbolic Algebra.* 13th IFAC Symposium on Control in Transportation Systems, The International Federation of Automatic Control, Sofia, Bulgaria, September, 2012.

[10] Michele Banci and Alessandro Fantechi. "Geographical Versus Functional Modelling by Statecharts of Interlocking Systems". In: *Electronic Notes in Theoretical Computer Science* 133 (2005), pp. 3–19.

[11] M. Banci, A. Fantechi and S. Gnesi. *The Role of Formal Methods in Developing a Distributed Railway Interlocking Systems.* Formal Methods & Tools Group, ISTI - CNR.

[12] Karim Kanso, Faron Moller and Anton Setzer. "Automated Verification of Signalling Principles in Railway Interlocking Systems". In: *Electronic Notes in Theoretical Computer Science* 250 (2009), pp. 19–31.

[13] Bidhan Malakar and Binoy Krishna Roy. "Railway Fail-Safe Signalization and Interlocking Design Based On Automation Petri Net". In: *ICICES2014 - S.A.Engineering College, Chennai, Tamil Nadu, India,* (2015). DOI: 10.1109/ICICES.2014.7034154. URL: https://www.researchgate.net/publication/281893403.

[14] Hugo Daniel dos Santos Macedo, Alessandro Fantechi and Anne Elisabeth Haxthausen. "Compositional Verification of Multi-Station Interlocking Systems". In: *Proceedings of the 7th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA 2016): Discussion, Dissemination, Applications - Part II* Springer. Lecture Notes in Computer Science, Vol.. 9953 (2016), pp. 279–293. DOI: https://doi.org/10.1007/978-3-319-47169-3_20.

[15] Eugenio Roanes-Lozano, Eugenio Roanes-Macias and Luis M. Laita. "A computer algebra approach to the design of routes and the study of their compatibility in a railway interlocking". In: *Mathematics and Computers in Simulation* 58 (2002), pp. 203–214.

[16] Wan Fokkink and Paul Hollingshead. *Verification of Interlockings: from Control Tables to Ladder Logic Diagrams.* University of Wales Swansea.

[17] Basri Tugcan Celebi and Ozgur Turay Kaymakci. "Verifying the accuracy of interlocking tables for railway signalling systems using abstract state machines". In: *J. Mod. Transport* 24(4) (2016), pp. 277–283. DOI: DOI10.1007/s40534-016-0119-1.

[18] Dong Wang, Xiangxian Chen and Hai Huang. "A graph theory-based approach to route location in railway interlocking". In: *Computers & Industrial Engineering* 66 (2013), pp. 791–799.

[19] Anne Elisabeth Haxthausen and Peter H. Østergaard. "On the Use of Static Checking in the Verification of Interlocking Systems". In: *Proceedings of the 7th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA 2016): Discussion, Dissemination, Applications - Part II* Springer. Lecture Notes in Computer Science, Vol.. 9953 (2016), pp. 266–278. DOI: https://doi.org/10.1007/978-3-319-47169-3_19.

[20] J. Berger, P. Middelraad and A.J. Smith. *EURIS, The European railway interlocking specification.* UIC, Commission 7A/16, 1992.

[21] Henrik Jonsson, Stig Larsson and Sasikumar Punnekkat. "Agile Practice in Regulated Railway Software Development". In: *IEEE 23rd International Symposium on Software Reliability Engineering Workshops* (2012).

[22] H.V. Vliet. *Software Engineering Principles and Practice.* John Wiley & Sons Ltd, England, Third edition, 2008.

[23] Linh Hong Vu and Anne Elisabeth Haxthausen. "Formal Development and Verification of Railway Control Systems - In the context of ERTMS/ETCS Level 2". In: *Technical University of Denmark (DTU)* (2015).

[24] Anne E. Haxthausen and Jan Peleska. "Formal Development and Verification of a Distributed Railway Control Systems". In: *IEEE Transactions on Software Engineering* 26 (2000), pp. 687–701.

[25] *EN 50128:2011 - Railway applications - Communications, signalling and processing systems - Software for railway control and protection systems.* CENELEC European Committee for Electrotechnical Standardization, April, 2014.

[26] Linh Hong Vu, Anne Elisabeth Haxthausen and Jan Peleska. "Formal modelling and verification of interlocking systems featuring sequential release". In: *Science of Computer Programming* 133 (2017), pp. 99–115. DOI: https://doi.org/10.1016/j.scico.2016.05.010.

[27] E. Roanes-Lozano and L.M. Laita. *An Approach from AI to the Design of Routes in a Railway Interlocking.* Dept. Algebra, Universidad Complutense de Madrid.

[28] M. Benerecetti et al. "Dynamic state machines for modelling railway control systems". In: *Science of Computer Programming* 133 (2017), pp. 116–153.

[29] D. Bjørner. "New results and trends in formal techniques and tools for the development of software for transportation systems: a review". In: *Proceedings 4th Symposium on Formal Methods for Railway Operation and Control Systems, FORMS03, LHarmattan Hongrie, Budapest* (2003).

[30] D. Harel. "Statecharts: a visual formalism for complex systems". In: *Science of Computer Programming* 8 (1987), pp. 231–274.

[31] OMG-UML2. *Unified Modeling Language: Infrastructure and Superstructure.* OMG, 2011, Version 2.4.1 formal/11-08-05.

[32] R. Alur, S. Kannan and M. Yannakakis. "Communicating hierarchical state machines, in: Automata, Languages and Programming". In: *Lect. Notes Comput. Sci., Springer, Berlin, Heidelberg* 1644 (1999), pp. 169–178.

[33] D. Harel and A. Naamad. "The statemate semantics of statecharts". In: *ACM Trans. Softw. Eng. Methodol* 5 (1996), pp. 293–333.

[34] Clarke E. "Birth of model checking". In: *25 years of model checking* 5000 (2008), pp. 1–26.

[35] Lichtenstein O and Pnueli A. "Checking that finite state concurrent programs satisfy their linear specification". In: *Proceedings of the 12th ACM SIGACT-SIGPLAN symposium on principles of programming languages* (1985). DOI: doi:10.1145/318593.318622.

[36] Wai Wong. *A Simple Graph Theory and Its Application in Railway Signalling.* Department of Engineering, University of Warwick, Coventry, England.

[37] *Rail Transport.* URL: http://traininfo.tech/rail-transport/. (accessed: 25.08.2020).

[38] L.T.C. Rolt. *Richard Trevithick, English engineer.* URL: https://www.britannica.com/biography/Richard-Trevithick. (accessed: 30.08.2020).

[39] SDR Signalling. *A Brief History of the Signalman & Signalling.* URL: http://sdrsignalling.com/History_of_Signalling.html. (accessed: 02.09.2020).

[40] Piers Connor. *Basic Railway Signalling.* Infopaper No. 6. Railway Technical Website, 2017.

[41] David Kerr and Tony Rowbotham. *Introduction to Railway Signalling.* Institution of Railway Signal Engineers, 2001.

[42] Trevor Moore. "Signalling Principles of Australian Rail Track Corporation". In: *IRSE NEWS* ISSUE 224 (2016).

[43] J. Maree et al. *Introduction to Multi-Disciplinary Concepts in Railway Engineering*. University of Pretoria, Chair in Railway Engineering, 2002.

[44] Yinghua Min et al. *A Fail-safe Microprocessor-based System for Interlocking on Railways*. Proceedings Annual Reliability and Maintainability Symposium, 1994.

[45] Kirsten Winter. *Optimising Ordering Strategies for Symbolic Model Checking of Railway Interlockings*. School of Information Technology and Electrical Engineering, The University of Queensland, St. Lucia, QLD 4072, Australia.

[46] Microsoft. *What is .NET Framework?* URL: `https://dotnet.microsoft.com/learn/dotnet/what-is-dotnet-framework`. (accessed: 25.01.2021).

[47] G. Michael Schneider and Judith L. Gersting. *Invitation to Computer Science - 7th edition*. Cengage Learning, 2016.

[48] Dr. Kieran Mulchrone. *An Introduction to Object Oriented Programming with C#*. Department of Applied Mathematics, University College, Cork., September 2010.

[49] Dan Clark. *Beginning C# Object-Oriented Programming - Second Edition*. Apress, Springer Science and Business Media, New York, 2013.

[50] Daisy Tang. *CS240 – Lecture Notes: Analysis of Algorithms*. URL: `https://www.cpp.edu/~ftang/courses/CS240/lectures/analysis.htm`. (accessed: 05.02.2021).

[51] Martin Escardo, Manfred Kerber and John Bullinaria. *Lecture Notes for Data Structures and Algorithms*. School of Computer Science, University of Birmingham, Birmingham, UK, March 2019.

[52] Peter Gacs and Laszlo Lovasz. *Complexity of Algorithms*. Lecture Notes, Boston University and Yale University, 1999.

[53] J Paul Gibson. *Complexity & Algorithm Analysis*. Lecture Note of MAT 7003 : Mathematical Foundations (for Software Engineering), 2012.

[54] CSE IIT. *Lesson 16: Class and Interaction Diagrams, Version 2*. Kharagpur.

[55] Joydip Kanjilal. *Association, aggregation, and composition in OOP explained*. URL: `https://www.infoworld.com/article/3029325/exploring-association-aggregation-and-composition-in-oop.html`. (accessed: 07.02.2021).

Appendix A

Program Generated Control Table

A.1 Introduction

This appendix illustrates all the signalling station layout whose control tables were generated automatically. There were a total of seven layouts from which four of those layouts were standard station layout that aided to identify the standard compositional sub-sections where as from the fifth to seventh layout were large and complex station layouts that contained the standard sub-sections. This appendix also presents the output text files of the auto-generated control table. All output files that were generated using the programs of conventional approach and compositional approach are documented.

A.2 Research Contents

The investigation was conducted by applying the auto-generating program for a number of station layouts using both, conventional and compositional, approaches. The work commenced with simpler standard station layout and then successively expanding the stations with more signalling elements to larger and more complex layout for auto-generating programs to output the control table accurately using both approaches. The following seven topology were simulated in this research work:

1. A Single Turnout Station
2. A Single Loop Station
3. A Single Crossover Station

4. A Double Crossover Station
5. A Complex Station
6. A Large Station
7. A Line Plan

The standard four station layouts are often found in the South African Railways. The identification of the signalling elements are adopted from South African Railways Signals, points and tracks numbering conventions.

Section A.3 to A.9 of this appendix display the above mentioned seven layouts for the conventional approach programs and also display clearly the boundaries of the sub-sections for the compositional approach. These sections also document the output files generated by the programs. The text files were generated for each station layout using both approaches.

There are three main parts of the control table, they are, route information, flank protection information and aspect information. The control table generator programs output three parts of the control table in three separate text files. First text file contains route information, second test file contains flank protection information and last text file contains aspect information.

A.3 Topology 1 - A Single Turnout Station

This is a typical station layout where a single mainline converts into two mainlines.

A.3.1 Layout Diagram

This topology in Figure A.1 shows the station layout with only one turnout point.

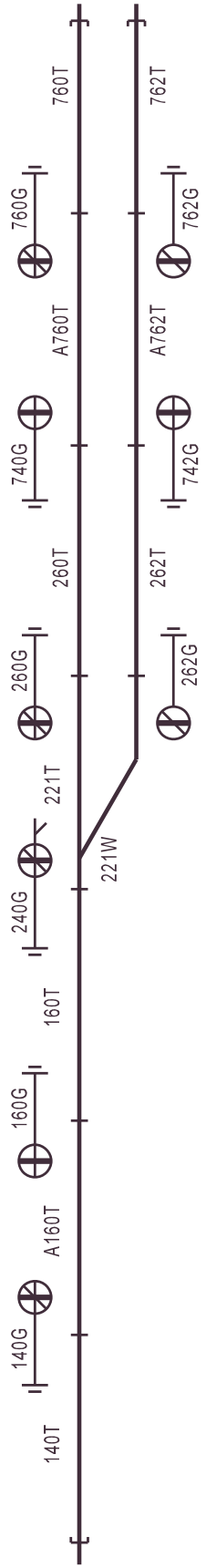


Figure A.1: A layout of the single turnout station

A.3.2 Output Files using Conventional Approach

The following text file is generated by the program using conventional approach which shows the route information.

Single_Turnout_Station_Route_Information(Conventional).txt

```
This file is generated using Control Table Generator program.
Generated on Saturday, 14 November 2020 22:49:55
-----------------------------------------------------------------------------------------------------

Topology: A Single Turnout Station Layout
Control Table Section: Route Information
Algorithm Design: Conventional Approach

Route      Departure    Destination   Tracks                Points Required          Points Required
Number     Signal       Signal        Required              In Straight Position     In Turnout Position
-----------------------------------------------------------------------------------------------------
1          140G         240G          A160T,160T
2          240G         740G          221T,260T             221W
3          240G         742G          221T,262T                                      221W
4          260G         160G          221T,160T             221W
5          760G         260G          A760T,260T
6          262G         160G          221T,160T                                      221W
7          762G         262G          A762T,262T
```

The following text file is generated by the program using conventional approach which shows the flank protection information.

Single_Turnout_Station_Flank_Information(Conventional).txt

```
This file is generated using Control Table Generator program.
Generated on Saturday, 14 November 2020 22:49:55
-----------------------------------------------------------------------------------------------------

Topology: A Single Turnout Station Layout
Control Table Section: Flank Protection Information
Algorithm Design: Conventional Approach

Route      Departure    Destination   Flank by              Flank by Points          Flank by Points
Number     Signal       Signal        Signals               In Straight Position     In Turnout Position
-----------------------------------------------------------------------------------------------------
1          140G         240G
2          240G         740G          262G
3          240G         742G          260G
4          260G         160G          262G
5          760G         260G
6          262G         160G          260G
7          762G         262G
```

The following text file is generated by the program using conventional approach which shows the aspect information.

Single_Turnout_Station_Aspect_Information(Conventional).txt

```
This file is generated using Control Table Generator program.
Generated on Saturday, 14 November 2020 22:49:55
-----------------------------------------------------------------------------------------------------

Topology: A Single Turnout Station Layout
Control Table Section: Aspect Information
Algorithm Design: Conventional Approach

Route      Departure    Destination   GREEN Aspect                  YELLOW Aspect
Number     Signal       Signal        Requirements                  Requirements
-----------------------------------------------------------------------------------------------------
1          140G         240G          240G GREEN,240G YELLOW        240G GREEN,240G YELLOW,240G RED
2          240G         740G          740G GREEN                    740G GREEN,740G RED
3          240G         742G                                        742G GREEN,742G RED
4          260G         160G          160G GREEN                    160G GREEN,160G RED
5          760G         260G          260G GREEN,260G YELLOW        260G GREEN,260G YELLOW,260G RED
6          262G         160G                                        160G GREEN,160G RED
7          762G         262G                                        262G YELLOW,262G RED
```

A.3.3 Layout Diagram with Compositional Cut-line

This topology in Figure A.2 shows the station layout with only one turnout point and clearly demarcates the compositional sub-sections.

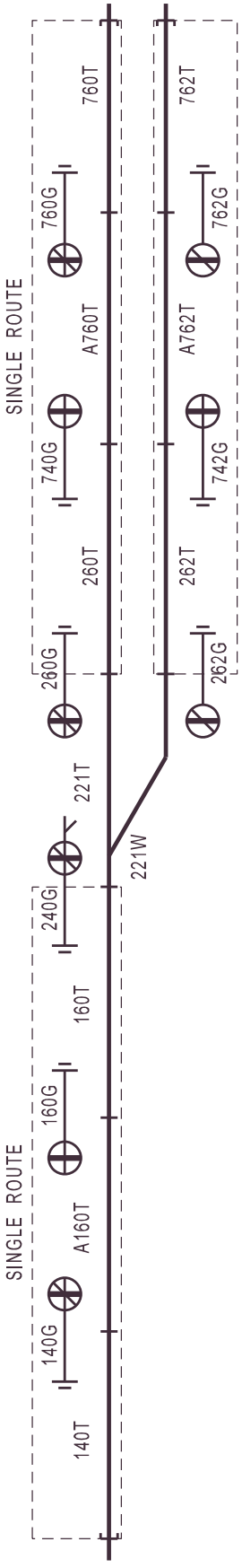


Figure A.2: A layout of the single turnout station with compositional cut-line

A.3.4 Output Files using Compositional Approach

The following text file is generated by the program using compositional approach which shows the route information.

Single_Turnout_Station_Route_Information(Compositional).txt

```
This file is generated using Control Table Generator program.
Generated on Sunday, 15 November 2020 21:38:25
-----------------------------------------------------------------------------------------------------

Topology: A Single Turnout Station Layout
Control Table Section: Route Information
Algorithm Design: Compositional Approach

Route      Departure    Destination   Tracks                Points Required         Points Required
Number     Signal       Signal        Required              In Straight Position    In Turnout Position
-----------------------------------------------------------------------------------------------------
1          140G         240G          A160T,160T
2          240G         740G          221T,260T             221W
3          240G         742G          221T,262T                                     221W
4          760G         260G          A760T,260T
5          260G         160G          221T,160T             221W
6          762G         262G          A762T,262T
7          262G         160G          221T,160T                                     221W
```

The following text file is generated by the program using compositional approach which shows the flank protection information.

Single_Turnout_Station_Flank_Information(Compositional).txt

```
This file is generated using Control Table Generator program.
Generated on Sunday, 15 November 2020 21:38:25
-----------------------------------------------------------------------------------------------------

Topology: A Single Turnout Station Layout
Control Table Section: Flank Protection Information
Algorithm Design: Compositional Approach

Route      Departure    Destination   Flank by              Flank by Points         Flank by Points
Number     Signal       Signal        Signals               In Straight Position    In Turnout Position
-----------------------------------------------------------------------------------------------------
1          140G         240G
2          240G         740G          262G
3          240G         742G          260G
4          760G         260G
5          260G         160G          262G
6          762G         262G
7          262G         160G          260G
```

The following text file is generated by the program using compositional approach which shows the aspect information.

Single_Turnout_Station_Aspect_Information(Compositional).txt

```
This file is generated using Control Table Generator program.
Generated on Sunday, 15 November 2020 21:38:25
-----------------------------------------------------------------------------------------------------

Topology: A Single Turnout Station Layout
Control Table Section: Aspect Information
Algorithm Design: Compositional Approach

Route      Departure    Destination   GREEN Aspect                  YELLOW Aspect
Number     Signal       Signal        Requirements                  Requirements
-----------------------------------------------------------------------------------------------------
1          140G         240G          240G GREEN,240G YELLOW        240G GREEN,240G YELLOW,240G RED
2          240G         740G          740G GREEN                    740G GREEN,740G RED
3          240G         742G                                        742G GREEN,742G RED
4          760G         260G          260G GREEN,260G YELLOW        260G GREEN,260G YELLOW,260G RED
5          260G         160G          160G GREEN                    160G GREEN,160G RED
6          762G         262G                                        262G YELLOW,262G RED
7          262G         160G                                        160G GREEN,160G RED
```

A.4 Topology 2 - A Single Loop Station

This is a typical station layout where there is only one loop in the station.

A.4.1 Layout Diagram

This topology in Figure A.3 shows the station layout with only one loop.

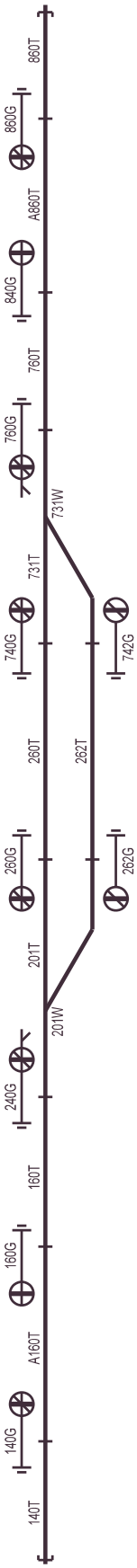


Figure A.3: A layout of the single loop station

A.4.2 Output Files using Conventional Approach

The following text file is generated by the program using conventional approach which shows the route information.

Single_Loop_Station_Route_Information(Conventional).txt

```
This file is generated using Control Table Generator program.
Generated on Sunday, 15 November 2020 23:41:20
-----------------------------------------------------------------------------------------------------

Topology: A Single Loop Station Layout
Control Table Section: Route Information
Algorithm Design: Conventional Approach

Route      Departure    Destination   Tracks                Points Required          Points Required
Number     Signal       Signal        Required              In Straight Position     In Turnout Position
-----------------------------------------------------------------------------------------------------
1          140G         240G          A160T,160T
2          240G         740G          201T,260T             201W
3          240G         742G          201T,262T                                      201W
4          260G         160G          201T,160T             201W
5          262G         160G          201T,160T                                      201W
6          740G         840G          201T,760T             731W
7          742G         840G          201T,760T                                      731W
8          760G         260G          201T,260T             731W
9          760G         262G          201T,262T                                      731W
10         860G         760G          A860T,760T
```

The following text file is generated by the program using conventional approach which shows the flank protection information.

Single_Loop_Station_Flank_Information(Conventional).txt

```
This file is generated using Control Table Generator program.
Generated on Sunday, 15 November 2020 23:41:20
-----------------------------------------------------------------------------------------------------

Topology: A Single Loop Station Layout
Control Table Section: Flank Protection Information
Algorithm Design: Conventional Approach

Route      Departure    Destination   Flank by              Flank by Points          Flank by Points
Number     Signal       Signal        Signals               In Straight Position     In Turnout Position
-----------------------------------------------------------------------------------------------------
1          140G         240G
2          240G         740G          262G
3          240G         742G          260G
4          260G         160G          262G
5          262G         160G          260G
6          740G         840G          742G
7          742G         840G          740G
```

```
8          760G          260G          742G
9          760G          262G          740G
10         860G          760G
```

The following text file is generated by the program using conventional approach which shows the aspect information.

Single_Loop_Station_Aspect_Information(Conventional).txt

```
This file is generated using Control Table Generator program.
Generated on Sunday, 15 November 2020 23:41:20
-------------------------------------------------------------------------------------------------------

Topology: A Single Loop Station Layout
Control Table Section: Aspect Information
Algorithm Design: Conventional Approach

Route      Departure    Destination   GREEN Aspect                  YELLOW Aspect
Number     Signal       Signal        Requirements                  Requirements
-------------------------------------------------------------------------------------------------------
1          140G         240G          240G GREEN,240G YELLOW        240G GREEN,240G YELLOW,240G RED
2          240G         740G          740G GREEN,740G YELLOW        740G GREEN,740G YELLOW,740G RED
3          240G         742G                                        742G YELLOW,742G RED
4          260G         160G          160G GREEN                    160G GREEN,160G RED
5          262G         160G                                        160G GREEN,160G RED
6          740G         840G          840G GREEN                    840G GREEN,840G RED
7          742G         840G                                        840G GREEN,840G RED
8          760G         260G          260G GREEN,260G YELLOW        260G GREEN,260G YELLOW,260G RED
9          760G         262G                                        262G YELLOW,262G RED
10         860G         760G          760G GREEN,760G YELLOW        760G GREEN,760G YELLOW,760G RED
```

A.4.3 Layout Diagram with Compositional Cut-line

This topology in Figure A.4 shows the station layout with only one Loop and clearly demarcates the compositional sub-sections.

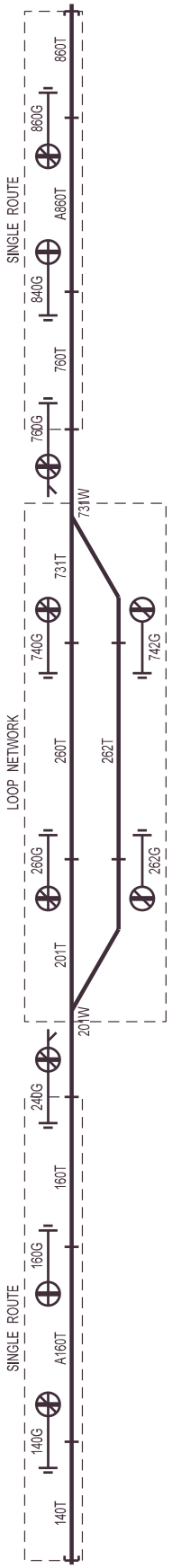


Figure A.4: A layout of the single loop station with compositional cut-line

A.4.4 Output Files using Compositional Approach

The following text file is generated by the program using compositional approach which shows the route information.

Single_Loop_Station_Route_Information(Compositional).txt

```
This file is generated using Control Table Generator program.
Generated on Monday, 16 November 2020 13:23:32
-----------------------------------------------------------------------------------------------------

Topology: A Single Loop Station Layout
Control Table Section: Route Information
Algorithm Design: Compositional Approach

Route      Departure    Destination   Tracks                Points Required          Points Required
Number     Signal       Signal        Required              In Straight Position     In Turnout Position
-----------------------------------------------------------------------------------------------------
1          140G         240G          A160T,160T
2          240G         740G          201T,260T             201W
3          240G         742G          201T,262T                                      201W
4          740G         840G          731T,760T             731W
5          742G         840G          731T,760T                                      731W
6          860G         760G          A860T,760T
7          760G         260G          731T,260T             731W
8          760G         262G          731T,262T                                      731W
9          260G         160G          201T,160T             201W
10         262G         160G          201T,160T                                      201W
```

The following text file is generated by the program using compositional approach which shows the flank protection information.

Single_Loop_Station_Flank_Information(Compositional).txt

```
This file is generated using Control Table Generator program.
Generated on Monday, 16 November 2020 13:23:32
-----------------------------------------------------------------------------------------------------

Topology: A Single Loop Station Layout
Control Table Section: Flank Protection Information
Algorithm Design: Compositional Approach

Route      Departure    Destination   Flank by              Flank by Points          Flank by Points
Number     Signal       Signal        Signals               In Straight Position     In Turnout Position
-----------------------------------------------------------------------------------------------------
1          140G         240G
2          240G         740G          262G
3          240G         742G          260G
4          740G         840G          742G
5          742G         840G          740G
6          860G         760G
7          760G         260G          742G
```

```
8          760G          262G          740G
9          260G          160G          262G
10         262G          160G          260G
```

The following text file is generated by the program using compositional approach which shows the aspect information.

Single_Loop_Station_Aspect_Information(Compositional).txt

```
This file is generated using Control Table Generator program.
Generated on Monday, 16 November 2020 13:23:32
-------------------------------------------------------------------------------------------------------

Topology: A Single Loop Station Layout
Control Table Section: Aspect Information
Algorithm Design: Compositional Approach

Route      Departure    Destination   GREEN Aspect                  YELLOW Aspect
Number     Signal       Signal        Requirements                  Requirements
-------------------------------------------------------------------------------------------------------
1          140G         240G          240G GREEN,240G YELLOW        240G GREEN,240G YELLOW,240G RED
2          240G         740G          740G GREEN,740G YELLOW        740G GREEN,740G YELLOW,740G RED
3          240G         742G                                        742G YELLOW,742G RED
4          740G         840G          840G GREEN                    840G GREEN,840G RED
5          742G         840G                                        840G GREEN,840G RED
6          860G         760G          760G GREEN,760G YELLOW        760G GREEN,760G YELLOW,760G RED
7          760G         260G          260G GREEN,260G YELLOW        260G GREEN,260G YELLOW,260G RED
8          760G         262G                                        262G YELLOW,262G RED
9          260G         160G          160G GREEN                    160G GREEN,160G RED
10         262G         160G                                        160G GREEN,160G RED
```

A.5 Topology 3 - A Single Crossover Station

This is a typical station layout where there are two mainlines and a crossover section to move from one line to another.

A.5.1 Layout Diagram

This topology in Figure A.5 shows the station layout with a single crossover.

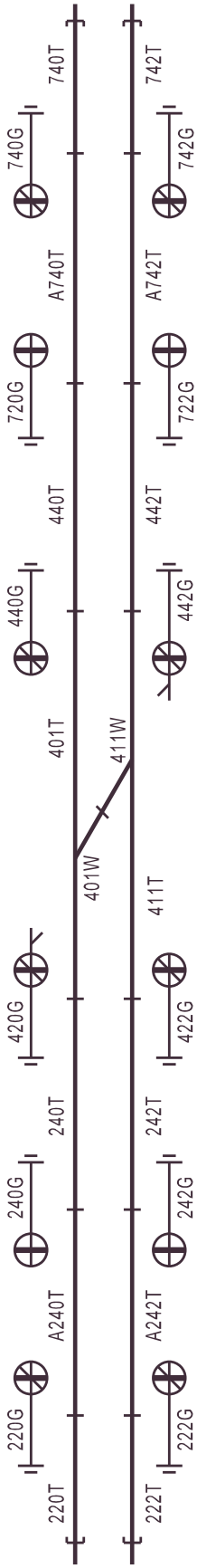


Figure A.5: A layout of the single crossover station

A.5.2 Output Files using Conventional Approach

The following text file is generated by the program using conventional approach which shows the route information.

Single_Crossover_Station_Route_Information(Conventional).txt

```
This file is generated using Control Table Generator program.
Generated on Monday, 16 November 2020 21:26:40
-------------------------------------------------------------------------------------------------------

Topology: A Single Crossover Station Layout
Control Table Section: Route Information
Algorithm Design: Conventional Approach

Route      Departure    Destination   Tracks                Points Required          Points Required
Number     Signal       Signal        Required              In Straight Position     In Turnout Position
-------------------------------------------------------------------------------------------------------
1          220G         420G          A240T,240T
2          222G         422G          A242T,242T
3          420G         720G          401T,440T             401W
4          420G         722G          401T,411T,442T                                 401W,411W
5          440G         240G          401T,240T             401W
6          422G         722G          411T,442T             411W
7          442G         242G          411T,242T             411W
8          442G         240G          411T,401T,240T                                 411W,401W
9          740G         440G          A740T,440T
10         742G         442G          A742T,442T
```

The following text file is generated by the program using conventional approach which shows the flank protection information.

Single_Crossover_Station_Flank_Information(Conventional).txt

```
This file is generated using Control Table Generator program.
Generated on Monday, 16 November 2020 21:26:40
-------------------------------------------------------------------------------------------------------

Topology: A Single Crossover Station Layout
Control Table Section: Flank Protection Information
Algorithm Design: Conventional Approach

Route      Departure    Destination   Flank by              Flank by Points          Flank by Points
Number     Signal       Signal        Signals               In Straight Position     In Turnout Position
-------------------------------------------------------------------------------------------------------
1          220G         420G
2          222G         422G
3          420G         720G                                411W
4          420G         722G          440G,422G
5          440G         240G                                411W
6          422G         722G                                401W
7          442G         242G                                401W
```

```
8          442G          240G          422G,440G
9          740G          440G
10         742G          442G
```

The following text file is generated by the program using conventional approach which shows the aspect information.

Single_Crossover_Station_Aspect_Information(Conventional).txt

```
This file is generated using Control Table Generator program.
Generated on Monday, 16 November 2020 21:26:40
-------------------------------------------------------------------------------------------------------

Topology: A Single Crossover Station Layout
Control Table Section: Aspect Information
Algorithm Design: Conventional Approach

Route      Departure    Destination   GREEN Aspect                  YELLOW Aspect
Number     Signal       Signal        Requirements                  Requirements
-------------------------------------------------------------------------------------------------------
1          220G         420G          420G GREEN,420G YELLOW        420G GREEN,420G YELLOW,420G RED
2          222G         422G          422G GREEN,422G YELLOW        422G GREEN,422G YELLOW,422G RED
3          420G         720G          720G GREEN                    720G GREEN,720G RED
4          420G         722G                                        722G GREEN,722G RED
5          440G         240G          240G GREEN                    240G GREEN,240G RED
6          422G         722G          722G GREEN                    722G GREEN,722G RED
7          442G         242G          242G GREEN                    242G GREEN,242G RED
8          442G         240G                                        240G GREEN,240G RED
9          740G         440G          440G GREEN,440G YELLOW        440G GREEN,440G YELLOW,440G RED
10         742G         442G          442G GREEN,442G YELLOW        442G GREEN,442G YELLOW,442G RED
```

A.5.3 Layout Diagram with Compositional Cut-line

This topology in Figure A.6 shows the station layout with a single crossover with cut-line.

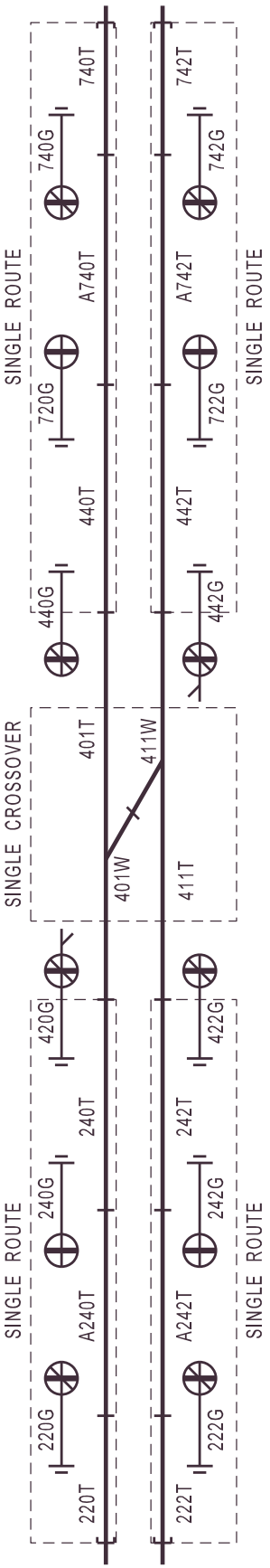


Figure A.6: A layout of the single crossover station with compositional cut-line

A.5.4 Output Files using Compositional Approach

The following text file is generated by the program using compositional approach which shows the route information.

Single_Crossover_Station_Route_Information(Compositional).txt

```
This file is generated using Control Table Generator program.
Generated on Monday, 16 November 2020 22:27:22
-----------------------------------------------------------------------------------------------------

Topology: A Single Crossover Station Layout
Control Table Section: Route Information
Algorithm Design: Compositional Approach

Route      Departure    Destination   Tracks                Points Required          Points Required
Number     Signal       Signal        Required              In Straight Position     In Turnout Position
-----------------------------------------------------------------------------------------------------
1          220G         420G          A240T,240T
2          420G         720G          401T,440T             401W
3          420G         722G          401T,411T,442T                                 401W,411W
4          740G         440G          A740T,440T
5          440G         240G          401T,240T             401W
6          222G         422G          A242T,242T
7          422G         722G          411T,442T             411W
8          742G         440G          A742T,442T
9          440G         242G          411T,242T             411W
10         440G         240G          411T,401T,240T                                 411W,401W
```

The following text file is generated by the program using compositional approach which shows the flank protection information.

Single_Crossover_Station_Flank_Information(Compositional).txt

```
This file is generated using Control Table Generator program.
Generated on Monday, 16 November 2020 22:27:22
-----------------------------------------------------------------------------------------------------

Topology: A Single Crossover Station Layout
Control Table Section: Flank Protection Information
Algorithm Design: Compositional Approach

Route      Departure    Destination   Flank by              Flank by Points          Flank by Points
Number     Signal       Signal        Signals               In Straight Position     In Turnout Position
-----------------------------------------------------------------------------------------------------
1          220G         420G
2          420G         720G                                411W
3          420G         722G          440G,422G
4          740G         440G
5          440G         240G                                411W
6          222G         422G
7          422G         722G                                401W
```

```
8          742G         440G
9          440G         242G                                401W
10         440G         240G          422G,440G
```

The following text file is generated by the program using compositional approach which shows the aspect information.

Single_Crossover_Station_Aspect_Information(Compositional).txt

```
This file is generated using Control Table Generator program.
Generated on Monday, 16 November 2020 22:27:22
-------------------------------------------------------------------------------------------------------

Topology: A Single Crossover Station Layout
Control Table Section: Aspect Information
Algorithm Design: Compositional Approach

Route      Departure    Destination   GREEN Aspect                  YELLOW Aspect
Number     Signal       Signal        Requirements                  Requirements
-------------------------------------------------------------------------------------------------------
1          220G         420G          420G GREEN,420G YELLOW        420G GREEN,420G YELLOW,420G RED
2          420G         720G          720G GREEN                    720G GREEN,720G RED
3          420G         722G                                        722G GREEN,722G RED
4          740G         440G          440G GREEN,440G YELLOW        440G GREEN,440G YELLOW,440G RED
5          440G         240G          240G GREEN                    240G GREEN,240G RED
6          222G         422G          422G GREEN,422G YELLOW        422G GREEN,422G YELLOW,422G RED
7          422G         722G          722G GREEN                    722G GREEN,722G RED
8          742G         440G          440G GREEN,440G YELLOW        440G GREEN,440G YELLOW,440G RED
9          440G         242G          242G GREEN                    242G GREEN,242G RED
10         440G         240G                                        240G GREEN,240G RED
```

A.6 Topology 4 - A Double Crossover Station

This is a typical station layout where there are two mainlines and a double crossover section to move from one line to another.

A.6.1 Layout Diagram

This topology in Figure A.7 shows the station layout with a double crossover.

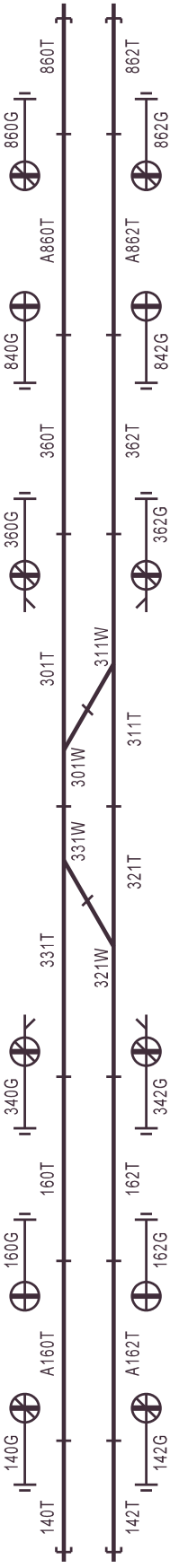


Figure A.7: A layout of the double crossover station

A.6.2 Output Files using Conventional Approach

The following text file is generated by the program using conventional approach which shows the route information.

Double_Crossover_Station_Route_Information(Conventional).txt

```
This file is generated using Control Table Generator program.
Generated on Monday, 16 November 2020 22:47:30
-----------------------------------------------------------------------------------------------------

Topology: A Double Crossover Station Layout
Control Table Section: Route Information
Algorithm Design: Conventional Approach

Route      Departure    Destination   Tracks                 Points Required          Points Required
Number     Signal       Signal        Required               In Straight Position     In Turnout Position
-----------------------------------------------------------------------------------------------------
1          140G         340G          A160T,160T
2          142G         342G          A162T,162T
3          340G         840G          331T,301T,360T         331W,301W
4          340G         842G          331T,301T,311T,362T    331W                     301W,311W
5          360G         160G          301T,331T,160T         301W,331W
6          360G         162G          301T,331T,321T,162T    301W                     331W,321W
7          342G         842G          321T,311T,362T         321W,311W
8          342G         840G          321T,331T,301T,360T    301W                     321W,331W
9          362G         162G          311T,321T,162T         311W,321W
10         362G         160G          311T,301T,331T,160T    331W                     311W,301W
11         860G         360G          A860T,360T
12         862G         362G          A862T,362T
```

The following text file is generated by the program using conventional approach which shows the flank protection information.

Double_Crossover_Station_Flank_Information(Conventional).txt

```
This file is generated using Control Table Generator program.
Generated on Monday, 16 November 2020 22:47:30
-----------------------------------------------------------------------------------------------------

Topology: A Double Crossover Station Layout
Control Table Section: Flank Protection Information
Algorithm Design: Conventional Approach

Route      Departure    Destination   Flank by               Flank by Points          Flank by Points
Number     Signal       Signal        Signals                In Straight Position     In Turnout Position
-----------------------------------------------------------------------------------------------------
1          140G         340G
2          142G         342G
3          340G         840G                                 321W,311W
4          340G         842G          342G,360G
5          360G         160G                                 311W,321W
```

```
6          360G          162G          362G,340G
7          342G          842G                                  331W,301W
8          342G          840G          362G,340G
9          362G          162G                                  301W,331W
10         362G          160G          342G,360G
11         860G          360G
12         862G          362G
```

The following text file is generated by the program using conventional approach which shows the aspect information.

Double_Crossover_Station_Aspect_Information(Conventional).txt

```
This file is generated using Control Table Generator program.
Generated on Monday, 16 November 2020 22:47:30
-------------------------------------------------------------------------------------------------------

Topology: A Double Crossover Station Layout
Control Table Section: Aspect Information
Algorithm Design: Conventional Approach

Route      Departure    Destination   GREEN Aspect                  YELLOW Aspect
Number     Signal       Signal        Requirements                  Requirements
-------------------------------------------------------------------------------------------------------
1          140G         340G          340G GREEN,340G YELLOW        340G GREEN,340G YELLOW,340G RED
2          142G         342G          342G GREEN,342G YELLOW        342G GREEN,342G YELLOW,342G RED
3          340G         840G          840G GREEN                    840G GREEN,840G RED
4          340G         842G                                        842G GREEN,842G RED
5          360G         160G          160G GREEN                    160G GREEN,160G RED
6          360G         162G                                        162G GREEN,162G RED
7          342G         842G          842G GREEN                    842G GREEN,842G RED
8          342G         840G                                        840G GREEN,840G RED
9          362G         162G          162G GREEN                    162G GREEN,162G RED
10         362G         160G                                        160G GREEN,160G RED
11         860G         360G          360G GREEN,360G YELLOW        360G GREEN,360G YELLOW,360G RED
12         862G         362G          362G GREEN,362G YELLOW        362G GREEN,362G YELLOW,362G RED
```

A.6.3 Layout Diagram with Compositional Cut-line

This topology in Figure A.8 shows the station layout with a double crossover with cut-line.

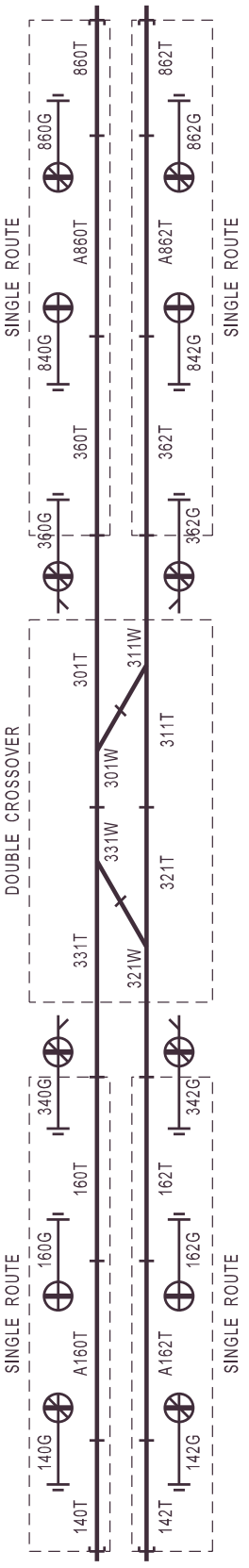


Figure A.8: A layout of the double crossover station with compositional cut-line

A.6.4 Output Files using Compositional Approach

The following text file is generated by the program using compositional approach which shows the route information.

Double_Crossover_Station_Route_Information(Compositional).txt

```
This file is generated using Control Table Generator program.
Generated on Monday, 16 November 2020 23:06:53
-----------------------------------------------------------------------------------------------------

Topology: A Double Crossover Station Layout
Control Table Section: Route Information
Algorithm Design: Compositional Approach

Route      Departure    Destination   Tracks                Points Required          Points Required
Number     Signal       Signal        Required              In Straight Position     In Turnout Position
-----------------------------------------------------------------------------------------------------
1          140G         340G          A160T,160T
2          340G         840G          331T,301T,360T        331W,301W
3          340G         842G          331T,301T,311T,362T   331W                     301W,311W
4          860G         360G          A860T,360T
5          360G         160G          301T,331T,160T        301W,331W
6          360G         162G          301T,331T,321T,162T   301W                     331W,321W
7          142G         342G          A162T,162T
8          342G         842G          321T,311T,362T        321W,311W
9          342G         840G          321T,331T,301T,360T   301W                     321W,331W
10         862G         362G          A862T,362T
11         362G         162G          311T,321T,162T        311W,321W
12         362G         160G          311T,301T,331T,160T   331W                     311W,301W
```

The following text file is generated by the program using compositional approach which shows the flank protection information.

Double_Crossover_Station_Flank_Information(Compositional).txt

```
This file is generated using Control Table Generator program.
Generated on Monday, 16 November 2020 23:06:53
-----------------------------------------------------------------------------------------------------

Topology: A Double Crossover Station Layout
Control Table Section: Flank Protection Information
Algorithm Design: Compositional Approach

Route      Departure    Destination   Flank by              Flank by Points          Flank by Points
Number     Signal       Signal        Signals               In Straight Position     In Turnout Position
-----------------------------------------------------------------------------------------------------
1          140G         340G
2          340G         840G                                321W,311W
3          340G         842G          360G,342G
4          860G         360G
5          360G         160G                                311W,321W
```

```
6          360G          162G          340G,362G
7          142G          342G
8          342G          842G                                  331W,301W
9          342G          840G          362G,340G
10         862G          362G
11         362G          162G                                  301W,331W
12         362G          160G          342G,360G
```

The following text file is generated by the program using compositional approach which shows the aspect information.

Double_Crossover_Station_Aspect_Information(Compositional).txt

```
This file is generated using Control Table Generator program.
Generated on Monday, 16 November 2020 23:06:53
-------------------------------------------------------------------------------------------------------

Topology: A Double Crossover Station Layout
Control Table Section: Aspect Information
Algorithm Design: Compositional Approach

Route      Departure    Destination   GREEN Aspect                  YELLOW Aspect
Number     Signal       Signal        Requirements                  Requirements
-------------------------------------------------------------------------------------------------------
1          140G         340G          340G GREEN,340G YELLOW        340G GREEN,340G YELLOW,340G RED
2          340G         840G          840G GREEN                    840G GREEN,840G RED
3          340G         842G                                        842G GREEN,842G RED
4          860G         360G          360G GREEN,360G YELLOW        360G GREEN,360G YELLOW,360G RED
5          360G         160G          160G GREEN                    160G GREEN,160G RED
6          360G         162G                                        162G GREEN,162G RED
7          142G         342G          342G GREEN,342G YELLOW        342G GREEN,342G YELLOW,342G RED
8          342G         842G          842G GREEN                    842G GREEN,842G RED
9          342G         840G                                        840G GREEN,840G RED
10         862G         362G          362G GREEN,362G YELLOW        362G GREEN,362G YELLOW,362G RED
11         362G         162G          162G GREEN                    162G GREEN,162G RED
12         362G         160G                                        160G GREEN,160G RED
```

A.7 Topology 5 - A Complex Station

This is a station layout of a complex station.

A.7.1 Layout Diagram

This topology in Figure A.9 shows a complex station layout.

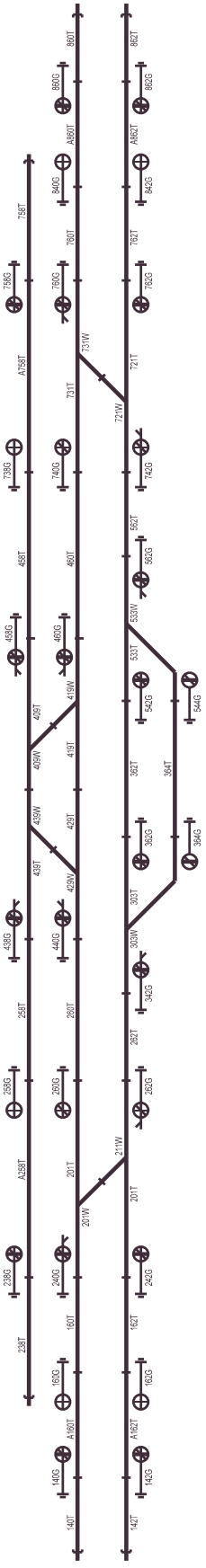


Figure A.9: A layout of the complex station

A.7.2 Output Files using Conventional Approach

The following text file is generated by the program using conventional approach which shows the route information.

Complex_Station_Route_Information(Conventional).txt

```
This file is generated using Control Table Generator program.
Generated on Monday, 16 November 2020 19:38:22
-------------------------------------------------------------------------------------------------------

Topology: A Complex Station Layout
Control Table Section: Route Information
Algorithm Design: Conventional Approach
```

Route Number	Departure Signal	Destination Signal	Tracks Required	Points Required In Straight Position	Points Required In Turnout Position
1	140G	240G	A160T,160T		
2	240G	440G	201T,260T	201W	
3	240G	342G	201T,211T,262T		201W,211W
4	142G	242G	A162T,162T		
5	242G	342G	211T,262T	211W	
6	260G	160G	201T,160T	201W	
7	262G	162G	211T,162T	211W	
8	262G	160G	211T,201T,160T		211W,201W
9	342G	542G	303T,362T	303W	
10	342G	544G	303T,364T		303W
11	362G	262G	303T,262T	303W	
12	364G	262G	303T,262T		303W
13	542G	742G	533T,562T	533W	
14	544G	742G	533T,562T		533W
15	562G	362G	533T,362T	533W	
16	562G	364G	533T,364T		533W
17	740G	840G	731T,760T	731W	
18	742G	842G	721T,762T	721W	
19	742G	840G	721T,731T,760T		721W,731W
20	760G	460G	731T,460T	731W	
21	760G	562G	731T,721T,562T		731W,721W
22	860G	760G	A860T,760T		
23	762G	562G	721T,562T	721W	
24	862G	762G	A862T,762T		
25	238G	438G	A258T,258T		
26	438G	738G	439T,409T,458T	439W,409W	
27	438G	740G	439T,409T,419T,460T	439W	409W,419W
28	440G	740G	429T,419T,460T	429W,419W	
29	440G	738G	429T,439T,409T,458T	409W	429W,439W
30	460G	260G	419T,429T,260T	419W,429W	
31	460G	258G	419T,409T,439T,258T	439W	419W,409W
32	458G	258G	409T,439T,258T	409W,439W	
33	458G	260G	409T,439T,429T,260T	409W	439W,429W
34	758G	458G	A758T,458T		

The following text file is generated by the program using conventional approach which

shows the flank protection information.

Complex_Station_Flank_Information(Conventional).txt

```
This file is generated using Control Table Generator program.
Generated on Monday, 16 November 2020 19:38:22
-----------------------------------------------------------------------------------------------------

Topology: A Complex Station Layout
Control Table Section: Flank Protection Information
Algorithm Design: Conventional Approach
```

Route Number	Departure Signal	Destination Signal	Flank by Signals	Flank by Points In Straight Position	Flank by Points In Turnout Position
1	140G	240G			
2	240G	440G		211W	
3	240G	342G	242G	439W	419W
4	142G	242G			
5	242G	342G		201W	
6	260G	160G		211W	
7	262G	162G		201W	
8	262G	160G	242G	439W	419W
9	342G	542G	364G		
10	342G	544G	362G		
11	362G	262G	364G		
12	364G	262G	362G		
13	542G	742G	544G		
14	544G	742G	542G		
15	562G	362G	544G		
16	562G	364G	542G		
17	740G	840G		721W	
18	742G	842G		731W	
19	742G	840G	762G	409W	429W
20	760G	460G		721W	
21	760G	562G	762G	409W	429W
22	860G	760G			
23	762G	562G		731W	
24	862G	762G			
25	238G	438G			
26	438G	738G		429W,419W	
27	438G	740G	458G		201W
28	440G	740G		439W,409W	
29	440G	738G	438G		731W
30	460G	260G		409W,439W	
31	460G	258G	458G		201W
32	458G	258G		419W,429W	
33	458G	260G	438G		731W
34	758G	458G			

The following text file is generated by the program using conventional approach which shows the aspect information.

Complex_Station_Aspect_Information(Conventional).txt

```
This file is generated using Control Table Generator program.
Generated on Monday, 16 November 2020 19:38:22
```

```
Topology: A Complex Station Layout
Control Table Section: Aspect Information
Algorithm Design: Conventional Approach
```

Route Number	Departure Signal	Destination Signal	GREEN Aspect Requirements	YELLOW Aspect Requirements
1	140G	240G	240G GREEN,240G YELLOW	240G GREEN,240G YELLOW,240G RED
2	240G	440G	440G GREEN,440G YELLOW	440G GREEN,440G YELLOW,440G RED
3	240G	342G		342G GREEN,342G YELLOW,342G RED
4	142G	242G	242G GREEN,242G YELLOW	242G GREEN,242G YELLOW,242G RED
5	242G	342G	342G GREEN,342G YELLOW	342G GREEN,342G YELLOW,342G RED
6	260G	160G	160G GREEN	160G GREEN,160G RED
7	262G	162G	162G GREEN	162G GREEN,162G RED
8	262G	160G		160G GREEN,160G RED
9	342G	542G	542G GREEN,542G YELLOW	542G GREEN,542G YELLOW,542G RED
10	342G	544G		544G YELLOW,544G RED
11	362G	262G	262G GREEN,262G YELLOW	262G GREEN,262G YELLOW,262G RED
12	364G	262G		262G GREEN,262G YELLOW,262G RED
13	542G	742G	742G GREEN,742G YELLOW	742G GREEN,742G YELLOW,742G RED
14	544G	742G		742G GREEN,742G YELLOW,742G RED
15	562G	362G	362G GREEN,362G YELLOW	362G GREEN,362G YELLOW,362G RED
16	562G	364G		364G YELLOW,364G RED
17	740G	840G	840G GREEN	840G GREEN,840G RED
18	742G	842G	842G GREEN	842G GREEN,842G RED
19	742G	840G		840G GREEN,840G RED
20	760G	460G	460G GREEN,460G YELLOW	460G GREEN,460G YELLOW,460G RED
21	760G	562G		562G GREEN,562G YELLOW,562G RED
22	860G	760G	760G GREEN,760G YELLOW	760G GREEN,760G YELLOW,760G RED
23	762G	562G	562G GREEN,562G YELLOW	562G GREEN,562G YELLOW,562G RED
24	862G	762G	762G GREEN,762G YELLOW	762G GREEN,762G YELLOW,762G RED
25	238G	438G	438G GREEN,438G YELLOW	438G GREEN,438G YELLOW,438G RED
26	438G	738G	738G GREEN	738G GREEN,738G RED
27	438G	740G		740G GREEN,740G YELLOW,740G RED
28	440G	740G	740G GREEN,740G YELLOW	740G GREEN,740G YELLOW,740G RED
29	440G	738G		738G GREEN,738G RED
30	460G	260G	260G GREEN,260G YELLOW	260G GREEN,260G YELLOW,260G RED
31	460G	258G		258G GREEN,258G RED
32	458G	258G	258G GREEN	258G GREEN,258G RED
33	458G	260G		260G GREEN,260G YELLOW,260G RED
34	758G	458G	458G GREEN,458G YELLOW	458G GREEN,458G YELLOW,458G RED

A.7.3 Layout Diagram with Compositional Cut-line

This topology in Figure A.10 shows the complex station layout with the with cut-line.

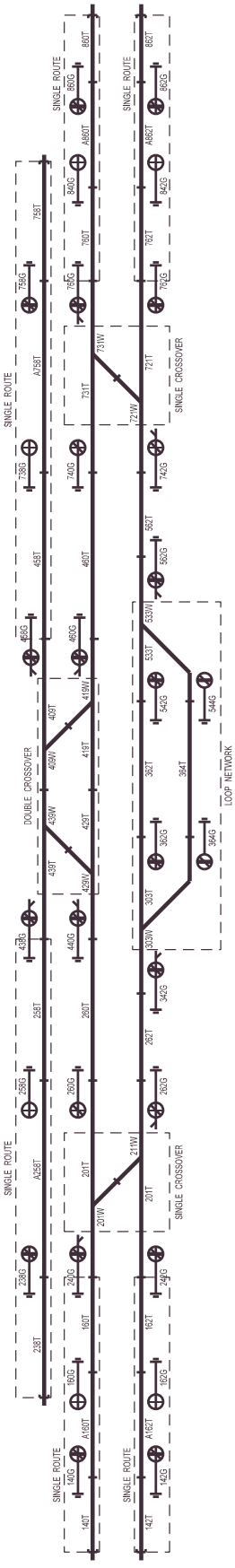


Figure A.10: A layout of the complex station with compositional cut-line

A.7.4 Output Files using Compositional Approach

The following text file is generated by the program using compositional approach which shows the route information.

Complex_Station_Route_Information(Compositional).txt

```
This file is generated using Control Table Generator program.
Generated on Monday, 16 November 2020 20:51:46
-----------------------------------------------------------------------------------------------------

Topology: A Complex Station Layout
Control Table Section: Route Information
Algorithm Design: Compositional Approach
```

Route Number	Departure Signal	Destination Signal	Tracks Required	Points Required In Straight Position	Points Required In Turnout Position
1	140G	240G	A160T,160T		
2	240G	440G	201T,260T	201W	
3	240G	342G	201T,211T,262T		201W,211W
4	142G	242G	A162T,162T		
5	242G	342G	211T,262T	211W	
6	238G	438G	A258T,258T		
7	438G	738G	439T,409T,458T	439W,409W	
8	438G	740G	439T,409T,419T,460T	439W	409W,419W
9	860G	760G	A860T,760T		
10	760G	460G	731T,460T	731W	
11	760G	562G	731T,721T,562T		731W,721W
12	862G	762G	A862T,762T		
13	762G	562G	721T,562T	721W	
14	758G	458G	A758T,458T		
15	458G	258G	409T,439T,258T	409W,439W	
16	458G	260G	409T,439T,429T,260T	409W	439W,429W
17	260G	160G	201T,160T	201W	
18	262G	162G	211T,162T	211W	
19	262G	160G	211T,201T,160T		211W,201W
20	440G	740G	429T,419T,460T	429W,419W	
21	440G	738G	429T,439T,409T,458T	409W	429W,439W
22	342G	542G	303T,362T	303W	
23	342G	544G	303T,364T		303W
24	542G	742G	533T,562T	533W	
25	544G	742G	533T,562T		533W
26	460G	260G	419T,429T,260T	419W,429W	
27	460G	258G	419T,409T,439T,258T	439W	419W,409W
28	562G	362G	533T,362T	533W	
29	562G	364G	533T,364T		533W
30	362G	262G	303T,262T	303W	
31	364G	262G	303T,262T		303W
32	740G	840G	731T,760T	731W	
33	742G	842G	721T,762T	721W	
34	742G	840G	721T,731T,760T		721W,731W

The following text file is generated by the program using compositional approach

which shows the flank protection information.

Complex_Station_Flank_Information(Compositional).txt

```
This file is generated using Control Table Generator program.
Generated on Monday, 16 November 2020 20:51:46
-------------------------------------------------------------------------------------------------------

Topology: A Complex Station Layout
Control Table Section: Flank Protection Information
Algorithm Design: Compositional Approach
```

Route Number	Departure Signal	Destination Signal	Flank by Signals	Flank by Points In Straight Position	Flank by Points In Turnout Position
1	140G	240G			
2	240G	440G		211W	
3	240G	342G	242G	439W	419W
4	142G	242G			
5	242G	342G		201W	
6	238G	438G			
7	438G	738G		429W,419W	
8	438G	740G	458G		201W
9	860G	760G			
10	760G	460G		721W	
11	760G	562G	762G	409W	429W
12	862G	762G			
13	762G	562G		731W	
14	758G	458G			
15	458G	258G		419W,429W	
16	458G	260G	438G		731W
17	260G	160G		211W	
18	262G	162G		201W	
19	262G	160G	242G	439W	419W
20	440G	740G		439W,409W	
21	440G	738G	438G		731W
22	342G	542G	364G		
23	342G	544G	362G		
24	542G	742G	544G		
25	544G	742G	542G		
26	460G	260G		409W,439W	
27	460G	258G	458G		201W
28	562G	362G	544G		
29	562G	364G	542G		
30	362G	262G	364G		
31	364G	262G	362G		
32	740G	840G		721W	
33	742G	842G		731W	
34	742G	840G	762G	409W	429W

The following text file is generated by the program using compositional approach which shows the aspect information.

Complex_Station_Aspect_Information(Compositional).txt

```
This file is generated using Control Table Generator program.
Generated on Monday, 16 November 2020 20:51:46
```

```
Topology: A Complex Station Layout
Control Table Section: Aspect Information
Algorithm Design: Compositional Approach
```

Route Number	Departure Signal	Destination Signal	GREEN Aspect Requirements	YELLOW Aspect Requirements
1	140G	240G	240G GREEN,240G YELLOW	240G GREEN,240G YELLOW,240G RED
2	240G	440G	440G GREEN,440G YELLOW	440G GREEN,440G YELLOW,440G RED
3	240G	342G		342G GREEN,342G YELLOW,342G RED
4	142G	242G	242G GREEN,242G YELLOW	242G GREEN,242G YELLOW,242G RED
5	242G	342G	342G GREEN,342G YELLOW	342G GREEN,342G YELLOW,342G RED
6	238G	438G	438G GREEN,438G YELLOW	438G GREEN,438G YELLOW,438G RED
7	438G	738G	738G GREEN	738G GREEN,738G RED
8	438G	740G		740G GREEN,740G YELLOW,740G RED
9	860G	760G	760G GREEN,760G YELLOW	760G GREEN,760G YELLOW,760G RED
10	760G	460G	460G GREEN,460G YELLOW	460G GREEN,460G YELLOW,460G RED
11	760G	562G		562G GREEN,562G YELLOW,562G RED
12	862G	762G	762G GREEN,762G YELLOW	762G GREEN,762G YELLOW,762G RED
13	762G	562G	562G GREEN,562G YELLOW	562G GREEN,562G YELLOW,562G RED
14	758G	458G	458G GREEN,458G YELLOW	458G GREEN,458G YELLOW,458G RED
15	458G	258G	258G GREEN	258G GREEN,258G RED
16	458G	260G		260G GREEN,260G YELLOW,260G RED
17	260G	160G	160G GREEN	160G GREEN,160G RED
18	262G	162G	162G GREEN	162G GREEN,162G RED
19	262G	160G		160G GREEN,160G RED
20	440G	740G	740G GREEN,740G YELLOW	740G GREEN,740G YELLOW,740G RED
21	440G	738G		738G GREEN,738G RED
22	342G	542G	542G GREEN,542G YELLOW	542G GREEN,542G YELLOW,542G RED
23	342G	544G		544G YELLOW,544G RED
24	542G	742G	742G GREEN,742G YELLOW	742G GREEN,742G YELLOW,742G RED
25	544G	742G		742G GREEN,742G YELLOW,742G RED
26	460G	260G	260G GREEN,260G YELLOW	260G GREEN,260G YELLOW,260G RED
27	460G	258G		258G GREEN,258G RED
28	562G	362G	362G GREEN,362G YELLOW	362G GREEN,362G YELLOW,362G RED
29	562G	364G		364G YELLOW,364G RED
30	362G	262G	262G GREEN,262G YELLOW	262G GREEN,262G YELLOW,262G RED
31	364G	262G		262G GREEN,262G YELLOW,262G RED
32	740G	840G	840G GREEN	840G GREEN,840G RED
33	742G	842G	842G GREEN	842G GREEN,842G RED
34	742G	840G		840G GREEN,840G RED

A.8 Topology 6 - A Large Station

This is a station layout of a large station that is much larger than the complex station shown in section A.7.

A.8.1 Layout Diagram

This topology in Figure A.11 shows a large station layout. The layout is too large to be able to draw in A4 page. Hence the layout is drawn in two parts with a joining line that denoted with X.

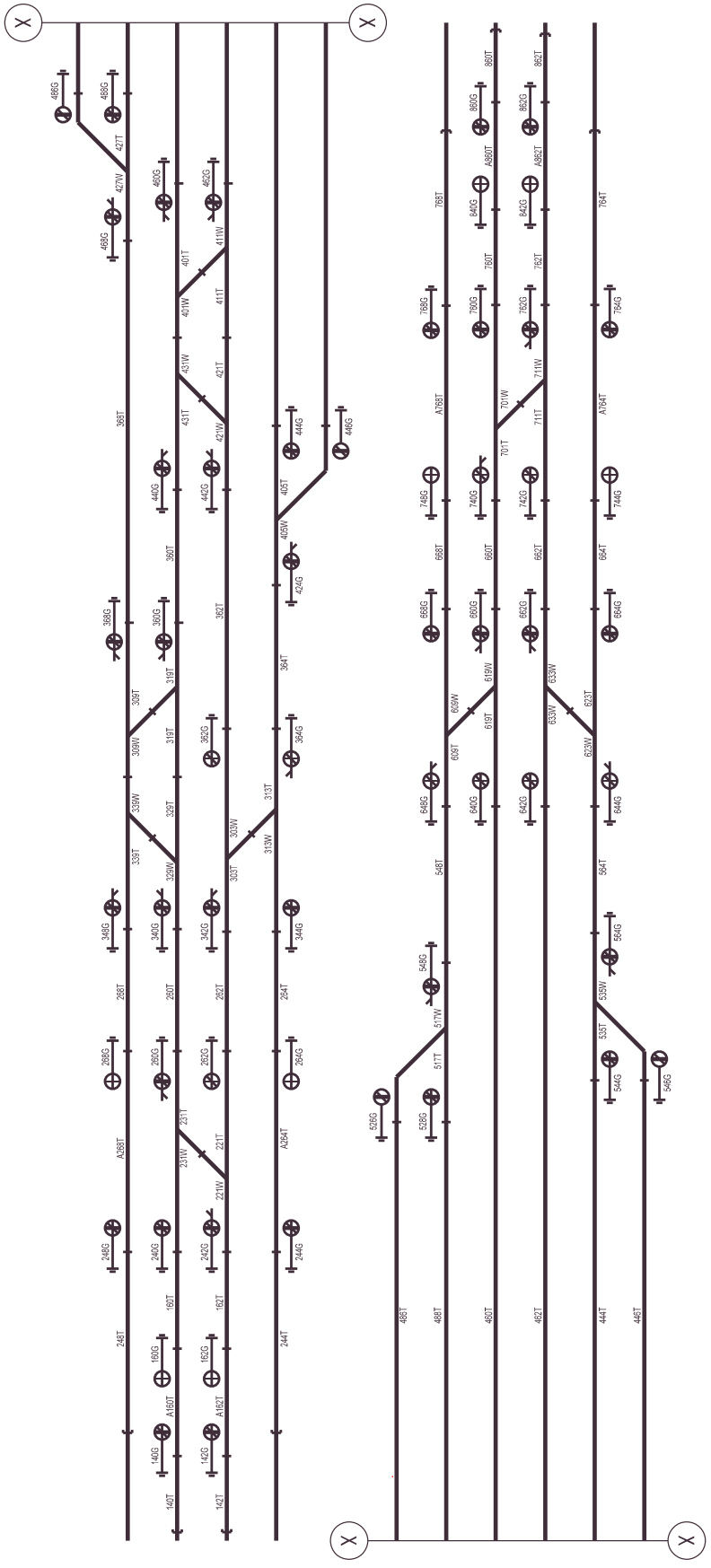


Figure A.11: A layout of the large station

A.8.2 Output Files using Conventional Approach

The following text file is generated by the program using conventional approach which shows the route information.

Large_Station_Route_Information(Conventional).txt

```
This file is generated using Control Table Generator program.
Generated on Thursday, 21 October 2021 12:07:00
```

Topology: A Large Station Layout
Control Table Section: Route Information
Algorithm Design: Conventional Approach

Route Number	Departure Signal	Destination Signal	Tracks Required	Points Required In Straight Position	Points Required In Turnout Position
1	140G	240G	A160T,160T		
2	240G	340G	231T,260T	231W	
3	142G	242G	A162T,162T		
4	242G	342G	221T,262T	221W	
5	242G	340G	221T,231T,260T		221W,231W
6	260G	160G	231T,160T	231W	
7	260G	162G	231T,221T,162T		231W,221W
8	262G	162G	221T,162T	221W	
9	248G	348G	A268T,268T		
10	244G	344G	A264T,264T		
11	348G	468G	339T,309T,368T	339W,309W	
12	348G	440G	339T,309T,319T,360T	339W	309W,319W
13	368G	268G	309T,339T,268T	309W,339W	
14	368G	260G	309T,339T,329T,260T	309W	339W,329W
15	340G	440G	329T,319T,360T	329W,319W	
16	340G	468G	329T,339T,309T,368T	309W	329W,339W
17	360G	260G	319T,329T,260T	319W,329W	
18	360G	268G	319T,309T,339T,268T	339W	319W,309W
19	342G	442G	303T,362T	303W	
20	342G	424G	303T,313T,364T		303W,313W
21	362G	262G	303T,262T	303W	
22	344G	424G	313T,364T	313W	
23	364G	264G	313T,264T	313W	
24	364G	262G	313T,303T,262T		313W,303W
25	424G	544G	405T,444T	405W	
26	424G	546G	405T,446T		405W
27	444G	364G	405T,364T	405W	
28	446G	364G	405T,364T		405W
29	440G	640G	431T,401T,460T	431W,401W	
30	440G	642G	431T,401T,411T,462T	431W	401W,411W
31	460G	360G	401T,431T,360T	401W,431W	
32	460G	362G	401T,431T,421T,362T	401W	431W,421W
33	442G	642G	421T,411T,462T	421W,411W	
34	442G	640G	421T,431T,401T,460T	401W	421W,431W
35	462G	362G	411T,421T,362T	411W,421W	
36	462G	360G	411T,401T,431T,360T	431W	411W,401W
37	468G	528G	427T,488T	427W	
38	468G	526G	427T,486T		427W
39	486G	368G	427T,368T		427W
40	488G	368G	427T,368T	427W	

41	526G	648G	517T,548T		517W
42	528G	648G	517T,548T	517W	
43	548G	488G	517T,488T	517W	
44	548G	486G	517T,486T		517W
45	544G	644G	535T,564T	535W	
46	564G	444G	535T,444T	535W	
47	564G	446G	535T,446T		535W
48	546G	644G	535T,564T		535W
49	648G	748G	609T,668T	609W	
50	648G	740G	609T,619T,660T		609W,619W
51	668G	548G	609T,548T	609W	
52	640G	740G	619T,660T	619W	
53	660G	460G	619T,460T	619W	
54	660G	548G	619T,609T,548T		619W,609W
55	642G	742G	633T,662T	633W	
56	662G	462G	633T,462T	633W	
57	662G	564G	633T,623T,564T		633W,623W
58	644G	744G	623T,664T	623W	
59	644G	742G	623T,633T,662T		623W,633W
60	664G	564G	623T,564T	623W	
61	768G	668G	A768T,668T		
62	740G	840G	701T,760T	701W	
63	740G	842G	701T,711T,762T		701W,711W
64	760G	660G	701T,660T	701W	
65	742G	842G	711T,762T	711W	
66	762G	662G	711T,662T	711W	
67	762G	660G	711T,701T,660T		711W,701W
68	764G	664G	A764T,664T		
69	860G	760G	A860T,760T		
70	862G	762G	A862T,762T		

The following text file is generated by the program using conventional approach which shows the flank protection information.

Large_Station_Flank_Information(Conventional).txt

```
This file is generated using Control Table Generator program.
Generated on Thursday, 21 October 2021 12:07:00
-----------------------------------------------------------------------------------------------

Topology: A Large Station Layout
Control Table Section: Flank Protection Information
Algorithm Design: Conventional Approach

Route      Departure    Destination   Flank by              Flank by Points         Flank by Points
Number     Signal       Signal        Signals               In Straight Position    In Turnout Position
-----------------------------------------------------------------------------------------------
1          140G         240G
2          240G         340G                                221W
3          142G         242G
4          242G         342G                                231W
5          242G         340G          240G                  313W,431W               411W,633W,711W
6          260G         160G                                221W
7          260G         162G          240G                  313W,431W               411W,633W,711W
```

8	262G	162G		231W	
9	248G	348G			
10	244G	344G			
11	348G	468G		329W,319W	
12	348G	440G	240G	221W,619W	
13	368G	268G		319W,329W	
14	368G	260G	348G		431W
15	340G	440G		339W,309W	
16	340G	468G	348G		431W
17	360G	260G		309W,339W	
18	360G	268G	240G	221W,619W	
19	342G	442G		313W	
20	342G	424G	344G	431W	411W
21	362G	262G		313W	
22	344G	424G		303W	
23	364G	264G		303W	
24	364G	262G	344G	431W	411W
25	424G	544G	446G		
26	424G	546G	444G		
27	444G	364G	446G		
28	446G	364G	444G		
29	440G	640G		421W,411W	
30	440G	642G			303W,619W
31	460G	360G		411W,421W	
32	460G	362G		309W,221W	633W,329W
33	442G	642G		431W,401W	
34	442G	640G		309W,221W	633W,329W
35	462G	362G		401W,431W	
36	462G	360G			303W,619W
37	468G	528G	486G		
38	468G	526G	488G		
39	486G	368G	488G		
40	488G	368G	486G		
41	526G	648G	528G		
42	528G	648G	526G		
43	548G	488G	526G		
44	548G	486G	528G		
45	544G	644G	546G		
46	564G	444G	546G		
47	564G	446G	544G		
48	546G	644G	544G		
49	648G	748G		619W	
50	648G	740G	668G		401W
51	668G	548G		619W	
52	640G	740G		609W	
53	660G	460G		609W	
54	660G	548G	668G		401W
55	642G	742G		623W	
56	662G	462G		623W	
57	662G	564G	664G	401W	421W
58	644G	744G		633W	
59	644G	742G	664G	401W	421W
60	664G	564G		633W	
61	768G	668G			
62	740G	840G		711W	
63	740G	842G	760G	623W	303W,221W
64	760G	660G		711W	
65	742G	842G		701W	
66	762G	662G		701W	
67	762G	660G	760G	623W	303W,221W
68	764G	664G			
69	860G	760G			
70	862G	762G			

The following text file is generated by the program using conventional approach which shows the aspect information.

Large_Station_Aspect_Information(Conventional).txt

```
This file is generated using Control Table Generator program.
Generated on Thursday, 21 October 2021 12:07:00
```

Topology: A Large Station Layout
Control Table Section: Aspect Information
Algorithm Design: Conventional Approach

Route Number	Departure Signal	Destination Signal	GREEN Aspect Requirements	YELLOW Aspect Requirements
1	140G	240G	240G GREEN,240G YELLOW	240G GREEN,240G YELLOW,240G RED
2	240G	340G	340G GREEN,340G YELLOW	340G GREEN,340G YELLOW,340G RED
3	142G	242G	242G GREEN,242G YELLOW	242G GREEN,242G YELLOW,242G RED
4	242G	342G	342G GREEN,342G YELLOW	342G GREEN,342G YELLOW,342G RED
5	242G	340G		340G GREEN,340G YELLOW,340G RED
6	260G	160G	160G GREEN	160G GREEN,160G RED
7	260G	162G		162G GREEN,162G RED
8	262G	162G	162G GREEN	162G GREEN,162G RED
9	248G	348G	348G GREEN,348G YELLOW	348G GREEN,348G YELLOW,348G RED
10	244G	344G	344G GREEN,344G YELLOW	344G GREEN,344G YELLOW,344G RED
11	348G	468G	468G GREEN,468G YELLOW	468G GREEN,468G YELLOW,468G RED
12	348G	440G		440G GREEN,440G YELLOW,440G RED
13	368G	268G	268G GREEN	268G GREEN,268G RED
14	368G	260G		260G GREEN,260G YELLOW,260G RED
15	340G	440G	440G GREEN,440G YELLOW	440G GREEN,440G YELLOW,440G RED
16	340G	468G		468G GREEN,468G YELLOW,468G RED
17	360G	260G	260G GREEN,260G YELLOW	260G GREEN,260G YELLOW,260G RED
18	360G	268G		268G GREEN,268G RED
19	342G	442G	442G GREEN,442G YELLOW	442G GREEN,442G YELLOW,442G RED
20	342G	424G		424G GREEN,424G YELLOW,424G RED
21	362G	262G	262G GREEN,262G YELLOW	262G GREEN,262G YELLOW,262G RED
22	344G	424G	424G GREEN,424G YELLOW	424G GREEN,424G YELLOW,424G RED
23	364G	264G	264G GREEN	264G GREEN,264G RED
24	364G	262G		262G GREEN,262G YELLOW,262G RED
25	424G	544G	544G GREEN,544G YELLOW	544G GREEN,544G YELLOW,544G RED
26	424G	546G		546G YELLOW,546G RED
27	444G	364G	364G GREEN,364G YELLOW	364G GREEN,364G YELLOW,364G RED
28	446G	364G		364G GREEN,364G YELLOW,364G RED
29	440G	640G	640G GREEN,640G YELLOW	640G GREEN,640G YELLOW,640G RED
30	440G	642G		642G GREEN,642G YELLOW,642G RED
31	460G	360G	360G GREEN,360G YELLOW	360G GREEN,360G YELLOW,360G RED
32	460G	362G		362G GREEN,362G YELLOW,362G RED
33	442G	642G	642G GREEN,642G YELLOW	642G GREEN,642G YELLOW,642G RED
34	442G	640G		640G GREEN,640G YELLOW,640G RED
35	462G	362G	362G GREEN,362G YELLOW	362G GREEN,362G YELLOW,362G RED
36	462G	360G		360G GREEN,360G YELLOW,360G RED
37	468G	528G	528G GREEN,528G YELLOW	528G GREEN,528G YELLOW,528G RED

38	468G	526G		526G YELLOW,526G RED
39	486G	368G		368G GREEN,368G YELLOW,368G RED
40	488G	368G	368G GREEN,368G YELLOW	368G GREEN,368G YELLOW,368G RED
41	526G	648G		648G GREEN,648G YELLOW,648G RED
42	528G	648G	648G GREEN,648G YELLOW	648G GREEN,648G YELLOW,648G RED
43	548G	488G	488G GREEN,488G YELLOW	488G GREEN,488G YELLOW,488G RED
44	548G	486G		486G YELLOW,486G RED
45	544G	644G	644G GREEN,644G YELLOW	644G GREEN,644G YELLOW,644G RED
46	564G	444G	444G GREEN,444G YELLOW	444G GREEN,444G YELLOW,444G RED
47	564G	446G		446G YELLOW,446G RED
48	546G	644G		644G GREEN,644G YELLOW,644G RED
49	648G	748G	748G GREEN	748G GREEN,748G RED
50	648G	740G		740G GREEN,740G YELLOW,740G RED
51	668G	548G	548G GREEN,548G YELLOW	548G GREEN,548G YELLOW,548G RED
52	640G	740G	740G GREEN,740G YELLOW	740G GREEN,740G YELLOW,740G RED
53	660G	460G	460G GREEN,460G YELLOW	460G GREEN,460G YELLOW,460G RED
54	660G	548G		548G GREEN,548G YELLOW,548G RED
55	642G	742G	742G GREEN,742G YELLOW	742G GREEN,742G YELLOW,742G RED
56	662G	462G	462G GREEN,462G YELLOW	462G GREEN,462G YELLOW,462G RED
57	662G	564G		564G GREEN,564G YELLOW,564G RED
58	644G	744G	744G GREEN	744G GREEN,744G RED
59	644G	742G		742G GREEN,742G YELLOW,742G RED
60	664G	564G	564G GREEN,564G YELLOW	564G GREEN,564G YELLOW,564G RED
61	768G	668G	668G GREEN,668G YELLOW	668G GREEN,668G YELLOW,668G RED
62	740G	840G	840G GREEN	840G GREEN,840G RED
63	740G	842G		842G GREEN,842G RED
64	760G	660G	660G GREEN,660G YELLOW	660G GREEN,660G YELLOW,660G RED
65	742G	842G	842G GREEN	842G GREEN,842G RED
66	762G	662G	662G GREEN,662G YELLOW	662G GREEN,662G YELLOW,662G RED
67	762G	660G		660G GREEN,660G YELLOW,660G RED
68	764G	664G	664G GREEN,664G YELLOW	664G GREEN,664G YELLOW,664G RED
69	860G	760G	760G GREEN,760G YELLOW	760G GREEN,760G YELLOW,760G RED
70	862G	762G	762G GREEN,762G YELLOW	762G GREEN,762G YELLOW,762G RED

A.8.3 Layout Diagram with Compositional Cut-line

This topology in Figure A.12 shows the complex station layout with the with cut-line.

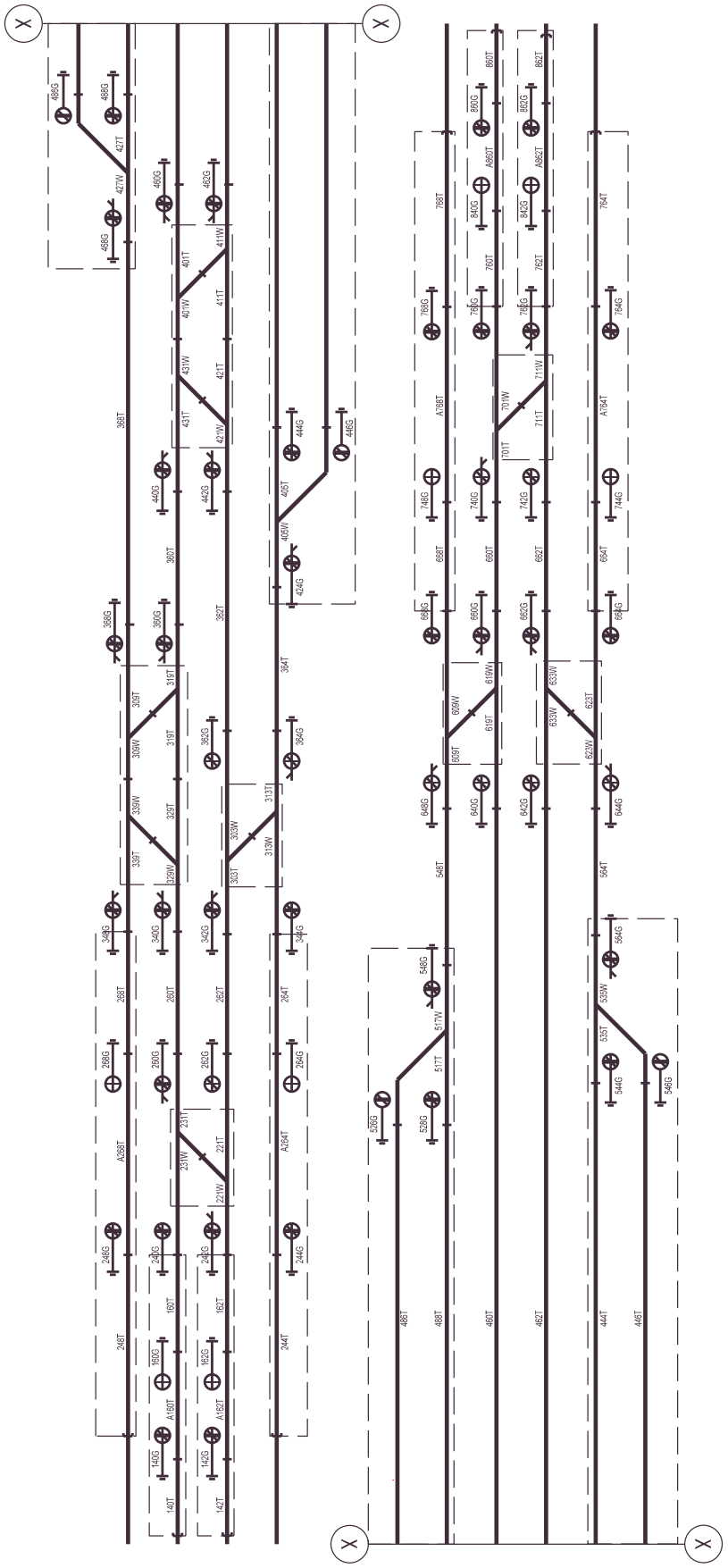


Figure A.12: A layout of the large station with compositional cut-line

A.8.4 Output Files using Compositional Approach

The following text file is generated by the program using compositional approach which shows the route information.

Large_Station_Route_Information(Compositional).txt

```
This file is generated using Control Table Generator program.
Generated on Thursday, 21 October 2021 12:12:51
```

Topology: A Large Station Layout
Control Table Section: Route Information
Algorithm Design: Compositional Approach

Route Number	Departure Signal	Destination Signal	Tracks Required	Points Required In Straight Position	Points Required In Turnout Position
1	248G	348G	A268T,268T		
2	348G	468G	339T,309T,368T	339W,309W	
3	348G	440G	339T,309T,319T,360T	339W	309W,319W
4	140G	240G	A160T,160T		
5	240G	340G	231T,260T	231W	
6	142G	242G	A162T,162T		
7	242G	342G	221T,262T	221W	
8	242G	340G	221T,231T,260T		221W,231W
9	244G	344G	A264T,264T		
10	344G	424G	313T,364T	313W	
11	768G	668G	A768T,668T		
12	668G	548G	609T,548T	609W	
13	860G	760G	A860T,760T		
14	760G	660G	701T,660T	701W	
15	862G	762G	A862T,762T		
16	762G	662G	711T,662T	711W	
17	762G	660G	711T,701T,660T		711W,701W
18	764G	664G	A764T,664T		
19	664G	564G	623T,564T	623W	
20	260G	160G	231T,160T	231W	
21	260G	162G	231T,221T,162T		231W,221W
22	340G	440G	329T,319T,360T	329W,319W	
23	340G	468G	329T,339T,309T,368T	309W	329W,339W
24	262G	162G	221T,162T	221W	
25	342G	442G	303T,362T	303W	
26	342G	424G	303T,313T,364T		303W,313W
27	368G	268G	309T,339T,268T	309W,339W	
28	368G	260G	309T,339T,329T,260T	309W	339W,329W
29	360G	260G	319T,329T,260T	319W,329W	
30	360G	268G	319T,309T,339T,268T	339W	319W,309W
31	362G	262G	303T,262T	303W	
32	364G	264G	313T,264T	313W	
33	364G	262G	313T,303T,262T		313W,303W
34	424G	544G	405T,444T	405W	
35	424G	546G	405T,446T		405W
36	544G	644G	535T,564T	535W	
37	546G	644G	535T,564T		535W
38	440G	640G	431T,401T,460T	431W,401W	
39	440G	642G	431T,401T,411T,462T	431W	401W,411W
40	442G	642G	421T,411T,462T	421W,411W	

41	442G	640G	421T,431T,401T,460T	401W	421W,431W
42	468G	528G	427T,488T	427W	
43	468G	526G	427T,486T		427W
44	528G	648G	517T,548T	517W	
45	526G	648G	517T,548T		517W
46	460G	360G	401T,431T,360T	401W,431W	
47	460G	362G	401T,431T,421T,362T	401W	431W,421W
48	462G	362G	411T,421T,362T	411W,421W	
49	462G	360G	411T,401T,431T,360T	431W	411W,401W
50	548G	488G	517T,488T	517W	
51	548G	486G	517T,486T		517W
52	488G	368G	427T,368T	427W	
53	486G	368G	427T,368T		427W
54	564G	444G	535T,444T	535W	
55	564G	446G	535T,446T		535W
56	444G	364G	405T,364T	405W	
57	446G	364G	405T,364T		405W
58	648G	748G	609T,668T	609W	
59	648G	740G	609T,619T,660T		609W,619W
60	640G	740G	619T,660T	619W	
61	642G	742G	633T,662T	633W	
62	644G	744G	623T,664T	623W	
63	644G	742G	623T,633T,662T		623W,633W
64	660G	460G	619T,460T	619W	
65	660G	548G	619T,609T,548T		619W,609W
66	662G	462G	633T,462T	633W	
67	662G	564G	633T,623T,564T		633W,623W
68	740G	840G	701T,760T	701W	
69	740G	842G	701T,711T,762T		701W,711W
70	742G	842G	711T,762T	711W	

The following text file is generated by the program using compositional approach which shows the flank protection information.

Large_Station_Flank_Information(Compositional).txt

```
This file is generated using Control Table Generator program.
Generated on Thursday, 21 October 2021 12:12:51
-----------------------------------------------------------------------------------------------

Topology: A Large Station Layout
Control Table Section: Flank Protection Information
Algorithm Design: Compositional Approach

Route      Departure    Destination   Flank by              Flank by Points         Flank by Points
Number     Signal       Signal        Signals               In Straight Position    In Turnout Position
-----------------------------------------------------------------------------------------------
1          248G         348G
2          348G         468G                                329W,319W
3          348G         440G          368G,340G
4          140G         240G
5          240G         340G                                221W
6          142G         242G
7          242G         342G                                231W
```

8	242G	340G	262G,240G			
9	244G	344G				
10	344G	424G		303W		
11	768G	668G				
12	668G	548G		619W		
13	860G	760G				
14	760G	660G		711W		
15	862G	762G				
16	762G	662G		701W		
17	762G	660G	742G,760G			
18	764G	664G				
19	664G	564G		633W		
20	260G	160G		221W		
21	260G	162G	240G,262G			
22	340G	440G		339W,309W		
23	340G	468G	348G			431W
24	262G	162G		231W		
25	342G	442G		313W		
26	342G	424G	344G			
27	368G	268G		319W,329W		
28	368G	260G	348G			431W
29	360G	260G		309W,339W		
30	360G	268G	340G,368G			
31	362G	262G		313W		
32	364G	264G		303W		
33	364G	262G	344G			
34	424G	544G	446G			
35	424G	546G	444G			
36	544G	644G	546G			
37	546G	644G	544G			
38	440G	640G		421W,411W		
39	440G	642G				303W
40	442G	642G		431W,401W		
41	442G	640G		309W		633W,329W
42	468G	528G	486G			
43	468G	526G	488G			
44	528G	648G	526G			
45	526G	648G	528G			
46	460G	360G		411W,421W		
47	460G	362G		309W		329W,633W
48	462G	362G		401W,431W		
49	462G	360G				303W
50	548G	488G	526G			
51	548G	486G	528G			
52	488G	368G	486G			
53	486G	368G	488G			
54	564G	444G	546G			
55	564G	446G	544G			
56	444G	364G	446G			
57	446G	364G	444G			
58	648G	748G		619W		
59	648G	740G	668G			401W
60	640G	740G		609W		
61	642G	742G		623W		
62	644G	744G		633W		
63	644G	742G	664G	401W		421W
64	660G	460G		609W		
65	660G	548G	668G			401W
66	662G	462G		623W		
67	662G	564G	664G	401W		421W
68	740G	840G		711W		
69	740G	842G	760G,742G			
70	742G	842G		701W		

The following text file is generated by the program using compositional approach which shows the aspect information.

Large_Station_Aspect_Information(Compositional).txt

```
This file is generated using Control Table Generator program.
Generated on Thursday, 21 October 2021 12:12:51
-----------------------------------------------------------------------------------------------------

Topology: A Large Station Layout
Control Table Section: Aspect Information
Algorithm Design: Compositional Approach
```

Route Number	Departure Signal	Destination Signal	GREEN Aspect Requirements	YELLOW Aspect Requirements
1	248G	348G	348G GREEN,348G YELLOW	348G GREEN,348G YELLOW,348G RED
2	348G	468G	468G GREEN,468G YELLOW	468G GREEN,468G YELLOW,468G RED
3	348G	440G		440G GREEN,440G YELLOW,440G RED
4	140G	240G	240G GREEN,240G YELLOW	240G GREEN,240G YELLOW,240G RED
5	240G	340G	340G GREEN,340G YELLOW	340G GREEN,340G YELLOW,340G RED
6	142G	242G	242G GREEN,242G YELLOW	242G GREEN,242G YELLOW,242G RED
7	242G	342G	342G GREEN,342G YELLOW	342G GREEN,342G YELLOW,342G RED
8	242G	340G		340G GREEN,340G YELLOW,340G RED
9	244G	344G	344G GREEN,344G YELLOW	344G GREEN,344G YELLOW,344G RED
10	344G	424G	424G GREEN,424G YELLOW	424G GREEN,424G YELLOW,424G RED
11	768G	668G	668G GREEN,668G YELLOW	668G GREEN,668G YELLOW,668G RED
12	668G	548G	548G GREEN,548G YELLOW	548G GREEN,548G YELLOW,548G RED
13	860G	760G	760G GREEN,760G YELLOW	760G GREEN,760G YELLOW,760G RED
14	760G	660G	660G GREEN,660G YELLOW	660G GREEN,660G YELLOW,660G RED
15	862G	762G	762G GREEN,762G YELLOW	762G GREEN,762G YELLOW,762G RED
16	762G	662G	662G GREEN,662G YELLOW	662G GREEN,662G YELLOW,662G RED
17	762G	660G		660G GREEN,660G YELLOW,660G RED
18	764G	664G	664G GREEN,664G YELLOW	664G GREEN,664G YELLOW,664G RED
19	664G	564G	564G GREEN,564G YELLOW	564G GREEN,564G YELLOW,564G RED
20	260G	160G	160G GREEN	160G GREEN,160G RED
21	260G	162G		162G GREEN,162G RED
22	340G	440G	440G GREEN,440G YELLOW	440G GREEN,440G YELLOW,440G RED
23	340G	468G		468G GREEN,468G YELLOW,468G RED
24	262G	162G	162G GREEN	162G GREEN,162G RED
25	342G	442G	442G GREEN,442G YELLOW	442G GREEN,442G YELLOW,442G RED
26	342G	424G		424G GREEN,424G YELLOW,424G RED
27	368G	268G	268G GREEN	268G GREEN,268G RED
28	368G	260G		260G GREEN,260G YELLOW,260G RED
29	360G	260G	260G GREEN,260G YELLOW	260G GREEN,260G YELLOW,260G RED
30	360G	268G		268G GREEN,268G RED
31	362G	262G	262G GREEN,262G YELLOW	262G GREEN,262G YELLOW,262G RED
32	364G	264G	264G GREEN	264G GREEN,264G RED
33	364G	262G		262G GREEN,262G YELLOW,262G RED
34	424G	544G	544G GREEN,544G YELLOW	544G GREEN,544G YELLOW,544G RED
35	424G	546G		546G YELLOW,546G RED
36	544G	644G	644G GREEN,644G YELLOW	644G GREEN,644G YELLOW,644G RED
37	546G	644G		644G GREEN,644G YELLOW,644G RED

38	440G	640G	640G GREEN,640G YELLOW	640G GREEN,640G YELLOW,640G RED
39	440G	642G		642G GREEN,642G YELLOW,642G RED
40	442G	642G	642G GREEN,642G YELLOW	642G GREEN,642G YELLOW,642G RED
41	442G	640G		640G GREEN,640G YELLOW,640G RED
42	468G	528G	528G GREEN,528G YELLOW	528G GREEN,528G YELLOW,528G RED
43	468G	526G		526G YELLOW,526G RED
44	528G	648G	648G GREEN,648G YELLOW	648G GREEN,648G YELLOW,648G RED
45	526G	648G		648G GREEN,648G YELLOW,648G RED
46	460G	360G	360G GREEN,360G YELLOW	360G GREEN,360G YELLOW,360G RED
47	460G	362G		362G GREEN,362G YELLOW,362G RED
48	462G	362G	362G GREEN,362G YELLOW	362G GREEN,362G YELLOW,362G RED
49	462G	360G		360G GREEN,360G YELLOW,360G RED
50	548G	488G	488G GREEN,488G YELLOW	488G GREEN,488G YELLOW,488G RED
51	548G	486G		486G YELLOW,486G RED
52	488G	368G	368G GREEN,368G YELLOW	368G GREEN,368G YELLOW,368G RED
53	486G	368G		368G GREEN,368G YELLOW,368G RED
54	564G	444G	444G GREEN,444G YELLOW	444G GREEN,444G YELLOW,444G RED
55	564G	446G		446G YELLOW,446G RED
56	444G	364G	364G GREEN,364G YELLOW	364G GREEN,364G YELLOW,364G RED
57	446G	364G		364G GREEN,364G YELLOW,364G RED
58	648G	748G	748G GREEN	748G GREEN,748G RED
59	648G	740G		740G GREEN,740G YELLOW,740G RED
60	640G	740G	740G GREEN,740G YELLOW	740G GREEN,740G YELLOW,740G RED
61	642G	742G	742G GREEN,742G YELLOW	742G GREEN,742G YELLOW,742G RED
62	644G	744G	744G GREEN	744G GREEN,744G RED
63	644G	742G		742G GREEN,742G YELLOW,742G RED
64	660G	460G	460G GREEN,460G YELLOW	460G GREEN,460G YELLOW,460G RED
65	660G	548G		548G GREEN,548G YELLOW,548G RED
66	662G	462G	462G GREEN,462G YELLOW	462G GREEN,462G YELLOW,462G RED
67	662G	564G		564G GREEN,564G YELLOW,564G RED
68	740G	840G	840G GREEN	840G GREEN,840G RED
69	740G	842G		842G GREEN,842G RED
70	742G	842G	842G GREEN	842G GREEN,842G RED

A.9 Topology 7 - A Line Plan

This is a layout of multiple stations of a railway network that starts with a single line and contains several loops and crossovers. The stations are usually labelled with their names. For this exercise the stations are labelled as station A to station H.

A.9.1 Layout Diagram

This topology in Figures A.13 to A.16 show the entire line plan of complete network.

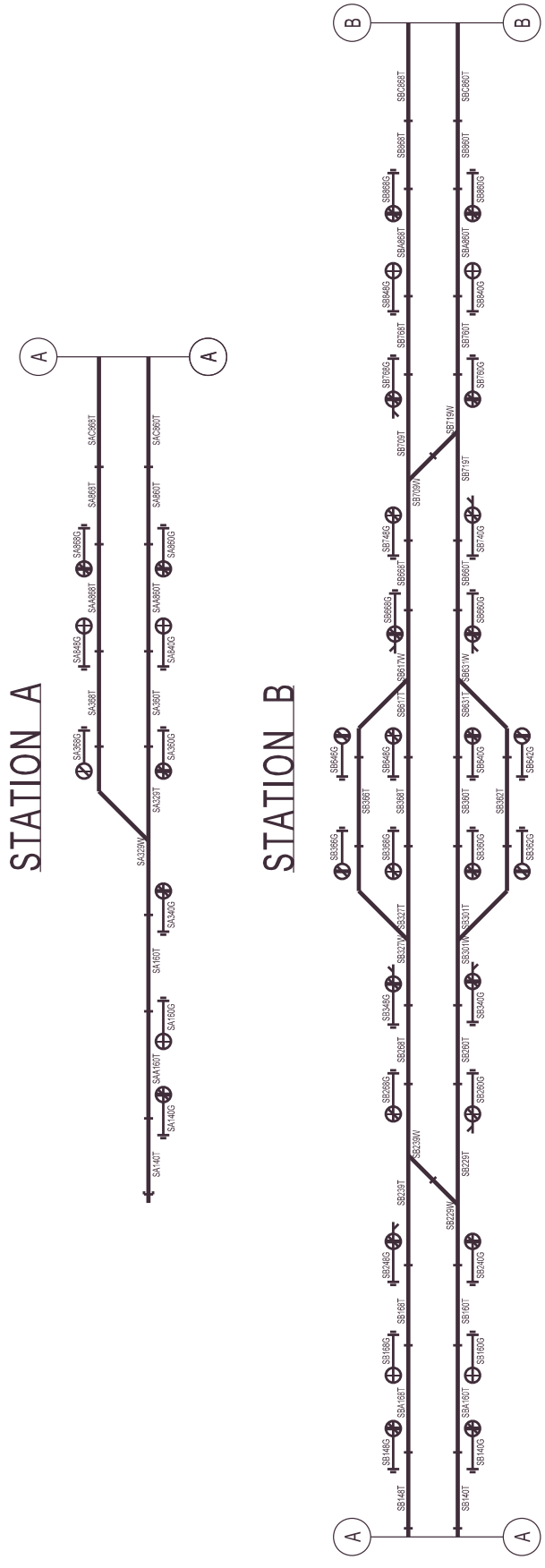


Figure A.13: The line plan (sheet 1)

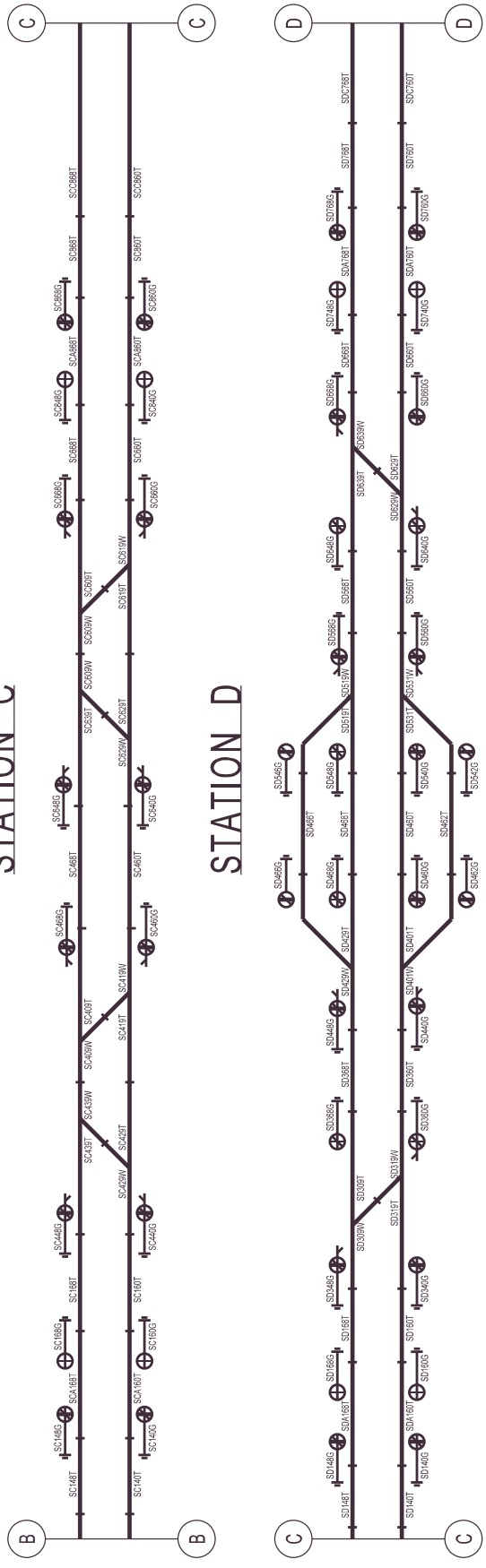


Figure A.14: The line plan (sheet 2)

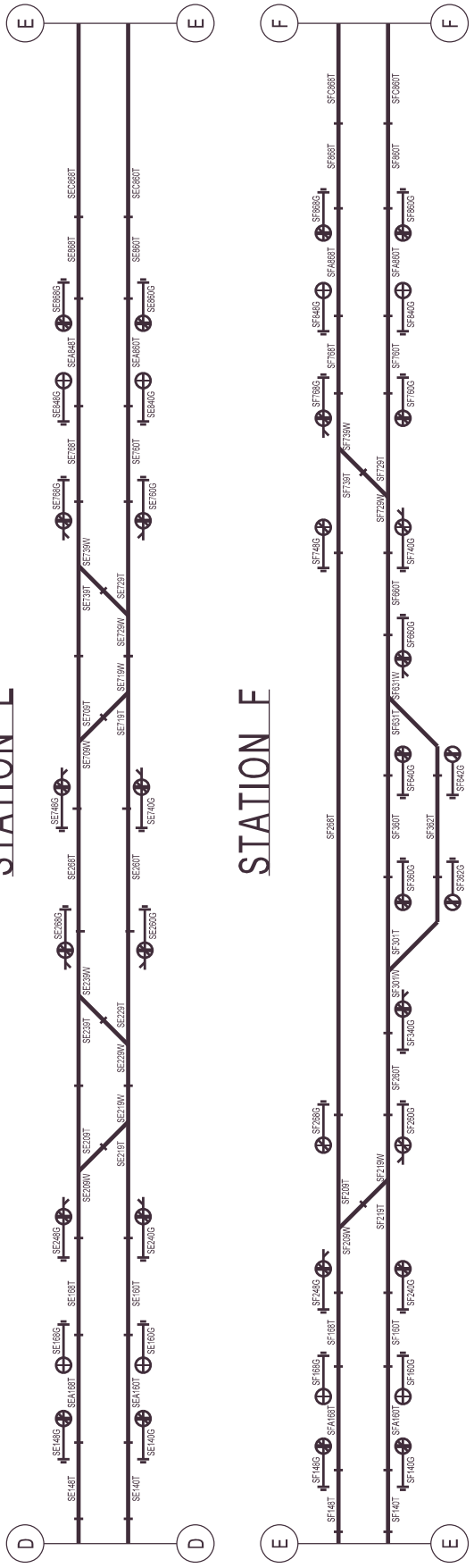


Figure A.15: The line plan (sheet 3)

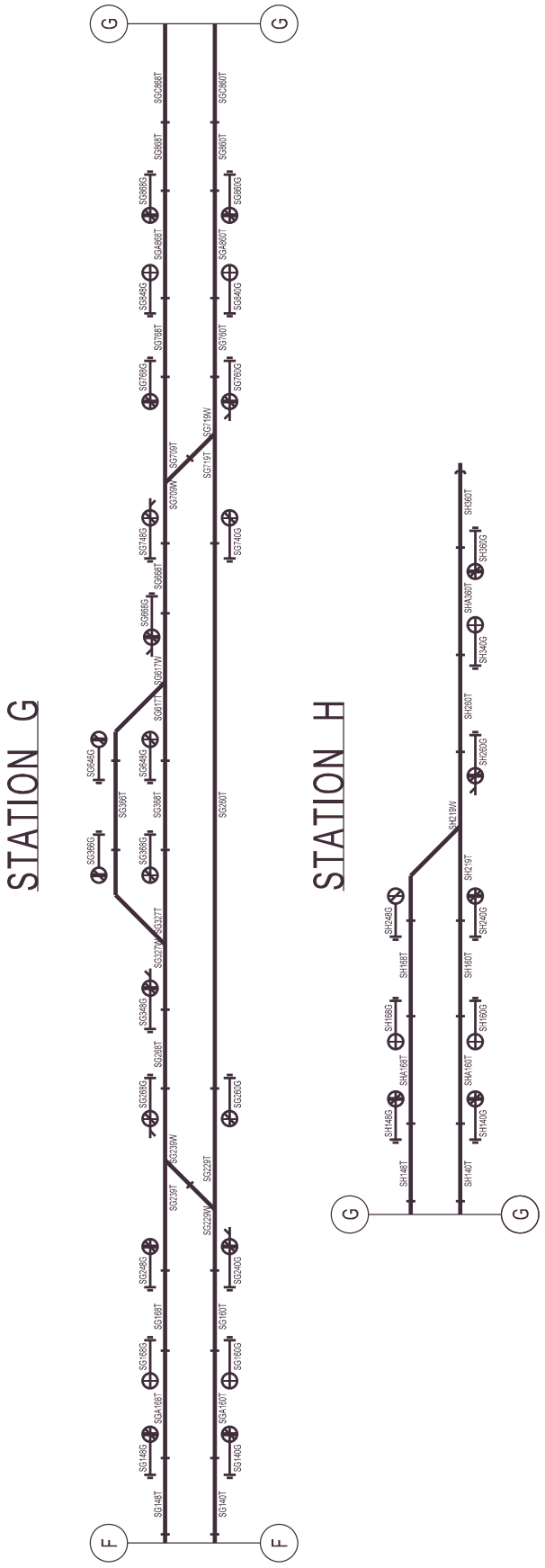


Figure A.16: The line plan (sheet 4)

A.9.2 Output Files using Conventional Approach

The following text file is generated by the program using conventional approach which shows the route information.

Line_Plan_Route_Information(Conventional).txt

```
This file is generated using Control Table Generator program.
Generated on Thursday, 14 October 2021 21:15:47
-----------------------------------------------------------------------------------------------------

Topology: A Line Plan
Control Table Section: Route Information
Algorithm Design: Conventional Approach
```

Route Number	Departure Signal	Destination Signal	Tracks Required	Points Required In Straight Position	Points Required In Turnout Position
1	SA140G	SA340G	SAA160T,SA160T		
2	SA340G	SA840G	SA329T,SA360T	SA329W	
3	SA340G	SA848G	SA329T,SA368T		SA329W
4	SA360G	SA160G	SA329T,SA160T	SA329W	
5	SA840G	SB140G	SAA860T,SA860T,SAC860T,SB140T		
6	SA860G	SA360G	SAA860T,SA360T		
7	SA368G	SA160G	SA329T,SA160T		SA329W
8	SA848G	SB148G	SAA868T,SA868T,SAC868T,SB148T		
9	SA868G	SA368G	SAA868T,SA368T		
10	SB148G	SB248G	SBA168T,SB168T		
11	SB168G	SA868G	SBA168T,SB148T,SAC868T,SA868T		
12	SB248G	SB348G	SB239T,SB268T	SB239W	
13	SB140G	SB240G	SBA160T,SB160T		
14	SB160G	SA860G	SBA160T,SB140T,SAC860T,SA860T		
15	SB240G	SB340G	SB229T,SB260T	SB229W	
16	SB240G	SB348G	SB229T,SB239T,SB268T		SB229W,SB239W
17	SB268G	SB168G	SB239T,SB168T	SB239W	
18	SB268G	SB160G	SB239T,SB229T,SB160T		SB239W,SB229W
19	SB260G	SB160G	SB229T,SB160T	SB229W	
20	SB348G	SB648G	SB327T,SB368T	SB327W	
21	SB348G	SB646G	SB327T,SB366T		SB327W
22	SB340G	SB640G	SB301T,SB360T	SB301W	
23	SB340G	SB642G	SB301T,SB362T		SB301W
24	SB366G	SB268G	SB327T,SB268T		SB327W
25	SB646G	SB748G	SB617T,SB668T		SB617W
26	SB368G	SB268G	SB327T,SB268T	SB327W	
27	SB648G	SB748G	SB617T,SB668T	SB617W	
28	SB360G	SB260G	SB301T,SB260T	SB301W	
29	SB640G	SB740G	SB631T,SB660T	SB631W	
30	SB362G	SB260G	SB301T,SB260T		SB301W
31	SB642G	SB740G	SB631T,SB660T		SB631W
32	SB668G	SB368G	SB617T,SB368T	SB617W	
33	SB668G	SB366G	SB617T,SB366T		SB617W
34	SB660G	SB360G	SB631T,SB360T	SB631W	
35	SB660G	SB362G	SB631T,SB362T		SB631W
36	SB748G	SB848G	SB709T,SB768T	SB709W	
37	SB748G	SB840G	SB709T,SB719T,SB760T		SB709W,SB719W
38	SB740G	SB840G	SB719T,SB760T	SB719W	
39	SB768G	SB668G	SB709T,SB668T	SB709W	
40	SB760G	SB660G	SB719T,SB660T	SB719W	

41	SB760G	SB668G	SB719T,SB709T,SB668T		SB719W,SB709W
42	SB848G	SC148G	SBA868T,SB868T,SBC868T,SC148T		
43	SB868G	SB768G	SBA868T,SB768T		
44	SB840G	SC140G	SBA860T,SB860T,SBC860T,SC140T		
45	SB860G	SB760G	SBA860T,SB760T		
46	SC148G	SC448G	SCA168T,SC168T		
47	SC168G	SB868G	SCA168T,SC148T,SBC868T,SB868T		
48	SC448G	SC648G	SC439T,SC409T,SC468T	SC439W,SC409W	
49	SC448G	SC640G	SC439T,SC409T,SC419T,SC460T	SC439W	SC409W,SC419W
50	SC140G	SC440G	SCA160T,SC160T		
51	SC160G	SB860G	SCA160T,SC140T,SBC860T,SB860T		
52	SC440G	SC640G	SC429T,SC419T,SC460T	SC429W,SC419W	
53	SC440G	SC648G	SC429T,SC439T,SC409T,SC468T	SC409W	SC429W,SC439W
54	SC468G	SC168G	SC409T,SC439T,SC168T	SC409W,SC439W	
55	SC468G	SC160G	SC409T,SC439T,SC429T,SC160T	SC409W	SC439W,SC429W
56	SC460G	SC160G	SC419T,SC429T,SC160T	SC419W,SC429W	
57	SC460G	SC168G	SC419T,SC409T,SC439T,SC168T	SC439W	SC419W,SC409W
58	SC648G	SC848G	SC639T,SC609T,SC668T	SC639W,SC609W	
59	SC648G	SC840G	SC639T,SC609T,SC619T,SC660T	SC639W	SC609W,SC619W
60	SC640G	SC840G	SC629T,SC619T,SC660T	SC629W,SC619W	
61	SC640G	SC848G	SC629T,SC639T,SC609T,SC668T	SC609W	SC629W,SC639W
62	SC668G	SC468G	SC609T,SC639T,SC468T	SC609W,SC639W	
63	SC668G	SC460G	SC609T,SC639T,SC629T,SC460T	SC609W	SC639W,SC629W
64	SC660G	SC460G	SC619T,SC629T,SC460T	SC619W,SC629W	
65	SC660G	SC468G	SC619T,SC609T,SC639T,SC468T	SC639W	SC619W,SC609W
66	SC848G	SD148G	SCA868T,SC868T,SCC868T,SD148T		
67	SC868G	SC668G	SCA868T,SC668T		
68	SC840G	SD140G	SCA860T,SC860T,SCC860T,SD140T		
69	SC860G	SC660G	SCA860T,SC660T		
70	SD148G	SD348G	SDA168T,SD168T		
71	SD168G	SC868G	SDA168T,SD148T,SCC868T,SC868T		
72	SD348G	SD448G	SD309T,SD368T	SD309W	
73	SD348G	SD440G	SD309T,SD319T,SD360T		SD309W,SD319W
74	SD140G	SD340G	SDA160T,SD160T		
75	SD160G	SC860G	SDA160T,SD140T,SCC860T,SC860T		
76	SD340G	SD440G	SD319T,SD360T	SD319W	
77	SD368G	SD168G	SD309T,SD168T	SD309W	
78	SD360G	SD160G	SD319T,SD160T	SD319W	
79	SD360G	SD168G	SD319T,SD309T,SD168T		SD319W,SD309W
80	SD448G	SD548G	SD429T,SD468T	SD429W	
81	SD448G	SD546G	SD429T,SD466T		SD429W
82	SD440G	SD540G	SD401T,SD460T	SD401W	
83	SD440G	SD542G	SD401T,SD462T		SD401W
84	SD466G	SD368G	SD429T,SD368T		SD429W
85	SD546G	SD648G	SD519T,SD568T		SD519W
86	SD468G	SD368G	SD429T,SD368T	SD429W	
87	SD548G	SD648G	SD519T,SD568T	SD519W	
88	SD460G	SD360G	SD401T,SD360T	SD401W	
89	SD540G	SD640G	SD531T,SD560T	SD531W	
90	SD462G	SD360G	SD401T,SD360T		SD401W
91	SD542G	SD640G	SD531T,SD560T		SD531W
92	SD568G	SD468G	SD519T,SD468T	SD519W	
93	SD568G	SD466G	SD519T,SD466T		SD519W
94	SD560G	SD460G	SD531T,SD460T	SD531W	
95	SD560G	SD462G	SD531T,SD462T		SD531W
96	SD648G	SD748G	SD639T,SD668T	SD639W	
97	SD640G	SD740G	SD629T,SD660T	SD629W	
98	SD640G	SD748G	SD629T,SD639T,SD668T		SD629W,SD639W
99	SD668G	SD568G	SD639T,SD568T	SD639W	
100	SD668G	SD560G	SD639T,SD629T,SD560T		SD639W,SD629W
101	SD660G	SD560G	SD629T,SD560T	SD629W	
102	SD748G	SE148G	SDA768T,SD768T,SDC768T,SE148T		
103	SD768G	SD668G	SDA768T,SD668T		

104	SD740G	SE140G	SDA760T,SD760T,SDC760T,SE140T		
105	SD760G	SD660G	SDA760T,SD660T		
106	SE148G	SE248G	SEA168T,SE168T		
107	SE168G	SD768G	SEA168T,SE148T,SDC768T,SD768T		
108	SE248G	SE748G	SE209T,SE239T,SE268T	SE209W,SE239W	
109	SE248G	SE740G	SE209T,SE219T,SE229T,SE260T	SE229W	SE209W,SE219W
110	SE140G	SE240G	SEA160T,SE160T		
111	SE160G	SD760G	SEA160T,SE140T,SDC760T,SD760T		
112	SE240G	SE740G	SE219T,SE229T,SE260T	SE219W,SE229W	
113	SE240G	SE748G	SE219T,SE229T,SE239T,SE268T	SE219W	SE229W,SE239W
114	SE268G	SE168G	SE239T,SE209T,SE168T	SE239W,SE209W	
115	SE268G	SE160G	SE239T,SE229T,SE219T,SE160T	SE219W	SE239W,SE229W
116	SE748G	SE848G	SE709T,SE739T,SE768T	SE709W,SE739W	
117	SE748G	SE840G	SE709T,SE719T,SE729T,SE760T	SE729W	SE709W,SE719W
118	SE260G	SE160G	SE229T,SE219T,SE160T	SE229W,SE219W	
119	SE260G	SE168G	SE229T,SE219T,SE209T,SE168T	SE229W	SE219W,SE209W
120	SE740G	SE840G	SE719T,SE729T,SE760T	SE719W,SE729W	
121	SE740G	SE848G	SE719T,SE729T,SE739T,SE768T	SE719W	SE729W,SE739W
122	SE768G	SE268G	SE739T,SE709T,SE268T	SE739W,SE709W	
123	SE768G	SE260G	SE739T,SE729T,SE719T,SE260T	SE719W	SE739W,SE729W
124	SE848G	SF148G	SEA868T,SE868T,SEC868T,SF148T		
125	SE868G	SE768G	SEA868T,SE768T		
126	SE760G	SE260G	SE729T,SE719T,SE260T	SE729W,SE719W	
127	SE760G	SE268G	SE729T,SE719T,SE709T,SE268T	SE729W	SE719W,SE709W
128	SE840G	SF140G	SEA860T,SE860T,SEC860T,SF140T		
129	SE860G	SE760G	SEA860T,SE760T		
130	SF148G	SF248G	SFA168T,SF168T		
131	SF168G	SE868G	SFA168T,SF148T,SEC868T,SE868T		
132	SF248G	SF748G	SF209T,SF268T	SF209W	
133	SF248G	SF340G	SF209T,SF219T,SF260T		SF209W,SF219W
134	SF140G	SF240G	SFA160T,SF160T		
135	SF160G	SE860G	SFA160T,SF140T,SEC860T,SE860T		
136	SF240G	SF340G	SF219T,SF260T	SF219W	
137	SF268G	SF168G	SF209T,SF168T	SF209W	
138	SF260G	SF160G	SF219T,SF160T	SF219W	
139	SF260G	SF168G	SF219T,SF209T,SF168T		SF219W,SF209W
140	SF340G	SF640G	SF301T,SF360T	SF301W	
141	SF340G	SF642G	SF301T,SF362T		SF301W
142	SF360G	SF260G	SF301T,SF260T	SF301W	
143	SF362G	SF260G	SF301T,SF260T		SF301W
144	SF660G	SF360G	SF631T,SF360T	SF631W	
145	SF660G	SF362G	SF631T,SF362T		SF631W
146	SF640G	SF740G	SF631T,SF660T	SF631W	
147	SF642G	SF740G	SF631T,SF660T		SF631W
148	SF748G	SF848G	SF739T,SF768T	SF739W	
149	SF740G	SF840G	SF729T,SF760T	SF729W	
150	SF740G	SF848G	SF729T,SF739T,SF768T		SF729W,SF739W
151	SF768G	SF268G	SF739T,SF268T	SF739W	
152	SF768G	SF660G	SF739T,SF729T,SF660T		SF739W,SF729W
153	SF760G	SF660G	SF729T,SF660T	SF729W	
154	SF848G	SG148G	SFA868T,SF868T,SFC868T,SG148T		
155	SF868G	SF768G	SFA868T,SF768T		
156	SF840G	SG140G	SFA860T,SF860T,SFC860T,SG140T		
157	SF860G	SF760G	SFA860T,SF760T		
158	SG148G	SG248G	SGA168T,SG168T		
159	SG168G	SF868G	SGA168T,SG148T,SFC868T,SF868T		
160	SG248G	SG348G	SG239T,SG268T	SG239W	
161	SG140G	SG240G	SGA160T,SG160T		
162	SG160G	SF860G	SGA160T,SG140T,SFC860T,SF860T		
163	SG240G	SG740G	SG229T,SG260T	SG229W	
164	SG240G	SG348G	SG229T,SG239T,SG268T		SG229W,SG239W
165	SG268G	SG168G	SG239T,SG168T	SG239W	
166	SG268G	SG160G	SG239T,SG229T,SG160T		SG239W,SG229W

167	SG260G	SG160G	SG229T,SG160T	SG229W	
168	SG348G	SG648G	SG327T,SG368T	SG327W	
169	SG348G	SG646G	SG327T,SG366T		SG327W
170	SG366G	SG268G	SG327T,SG268T		SG327W
171	SG646G	SG748G	SG617T,SG668T		SG617W
172	SG368G	SG268G	SG327T,SG268T	SG327W	
173	SG648G	SG748G	SG617T,SG668T	SG617W	
174	SG668G	SG368G	SG617T,SG368T	SG617W	
175	SG668G	SG366G	SG617T,SG366T		SG617W
176	SG748G	SG848G	SG709T,SG768T	SG709W	
177	SG748G	SG840G	SG709T,SG719T,SG760T		SG709W,SG719W
178	SG740G	SG840G	SG719T,SG760T	SG719W	
179	SG768G	SG668G	SG709T,SG668T	SG709W	
180	SG760G	SG260G	SG719T,SG260T	SG719W	
181	SG760G	SG668G	SG719T,SG709T,SG668T		SG719W,SG709W
182	SG848G	SH148G	SGA868T,SG868T,SGC868T,SH148T		
183	SG868G	SG768G	SGA868T,SG768T		
184	SG840G	SH140G	SGA860T,SG860T,SGC860T,SH140T		
185	SG860G	SG760G	SGA860T,SG760T		
186	SH148G	SH248G	SHA168T,SH168T		
187	SH168G	SG868G	SHA168T,SH148T,SGC868T,SG868T		
188	SH248G	SH340G	SH219T,SH260T		SH219W
189	SH140G	SH240G	SHA160T,SH160T		
190	SH160G	SG860G	SHA160T,SH140T,SGC860T,SG860T		
191	SH240G	SH340G	SH219T,SH260T	SH219W	
192	SH260G	SH160G	SH219T,SH160T	SH219W	
193	SH260G	SH168G	SH219T,SH168T		SH219W
194	SH360G	SH260G	SHA360T,SH260T		

The following text file is generated by the program using conventional approach which shows the flank protection information.

Line_Plan_Flank_Information(Conventional).txt

```
This file is generated using Control Table Generator program.
Generated on Thursday, 14 October 2021 21:15:47
-----------------------------------------------------------------------------------------------------

Topology: A Line Plan
Control Table Section: Flank Protection Information
Algorithm Design: Conventional Approach

Route      Departure    Destination   Flank by               Flank by Points         Flank by Points
Number     Signal       Signal        Signals                In Straight Position    In Turnout Position
-----------------------------------------------------------------------------------------------------
1          SA140G       SA340G
2          SA340G       SA840G                                                       SB239W
3          SA340G       SA848G                               SB239W,SB631W           SB631W
4          SA360G       SA160G                                                       SB239W
5          SA840G       SB140G
6          SA860G       SA360G
7          SA368G       SA160G                               SB239W,SB631W           SB631W
8          SA848G       SB148G
9          SA868G       SA368G
```

10	SB148G	SB248G				
11	SB168G	SA868G				
12	SB248G	SB348G		SB229W		
13	SB140G	SB240G				
14	SB160G	SA860G				
15	SB240G	SB340G		SB239W		
16	SB240G	SB348G		SA329W		
17	SB268G	SB168G		SB229W		
18	SB268G	SB160G		SA329W		
19	SB260G	SB160G		SB239W		
20	SB348G	SB648G	SB366G			
21	SB348G	SB646G	SB368G			
22	SB340G	SB640G	SB362G			
23	SB340G	SB642G	SB360G			
24	SB366G	SB268G	SB368G			
25	SB646G	SB748G	SB648G			
26	SB368G	SB268G	SB366G			
27	SB648G	SB748G	SB646G			
28	SB360G	SB260G	SB362G			
29	SB640G	SB740G	SB642G			
30	SB362G	SB260G	SB360G			
31	SB642G	SB740G	SB640G			
32	SB668G	SB368G	SB646G			
33	SB668G	SB366G	SB648G			
34	SB660G	SB360G	SB642G			
35	SB660G	SB362G	SB640G			
36	SB748G	SB848G		SB719W		
37	SB748G	SB840G			SC439W	
38	SB740G	SB840G		SB709W		
39	SB768G	SB668G		SB719W		
40	SB760G	SB660G		SB709W		
41	SB760G	SB668G			SC439W	
42	SB848G	SC148G				
43	SB868G	SB768G				
44	SB840G	SC140G				
45	SB860G	SB760G				
46	SC148G	SC448G				
47	SC168G	SB868G				
48	SC448G	SC648G		SC429W,SC419W		
49	SC448G	SC640G	SC468G	SB709W,SB301W	SB301W	
50	SC140G	SC440G				
51	SC160G	SB860G				
52	SC440G	SC640G		SC439W,SC409W		
53	SC440G	SC648G		SE219W	SE239W,SE219W,SB709W	
54	SC468G	SC168G		SC419W,SC429W		
55	SC468G	SC160G		SE219W	SE239W,SE219W,SB709W	
56	SC460G	SC160G		SC409W,SC439W		
57	SC460G	SC168G	SC468G	SB709W,SB301W	SB301W	
58	SC648G	SC848G		SC629W,SC619W		
59	SC648G	SC840G		SB709W,SB301W,SE219W	SB301W,SE239W	
60	SC640G	SC840G		SC639W,SC609W		
61	SC640G	SC848G	SC660G,SC648G			
62	SC668G	SC468G		SC619W,SC629W		
63	SC668G	SC460G	SC660G,SC648G			
64	SC660G	SC460G		SC609W,SC639W		
65	SC660G	SC468G		SB709W,SB301W,SE219W	SB301W,SE239W	
66	SC848G	SD148G				
67	SC868G	SC668G				
68	SC840G	SD140G				
69	SC860G	SC660G				
70	SD148G	SD348G				
71	SD168G	SC868G				
72	SD348G	SD448G		SD319W		

73	SD348G	SD440G		SB709W,SB301W	SB301W,SB709W
74	SD140G	SD340G			
75	SD160G	SC860G			
76	SD340G	SD440G		SD309W	
77	SD368G	SD168G		SD319W	
78	SD360G	SD160G		SD309W	
79	SD360G	SD168G		SB709W,SB301W	SB301W,SB709W
80	SD448G	SD548G	SD466G		
81	SD448G	SD546G	SD468G		
82	SD440G	SD540G	SD462G		
83	SD440G	SD542G	SD460G		
84	SD466G	SD368G	SD468G		
85	SD546G	SD648G	SD548G		
86	SD468G	SD368G	SD466G		
87	SD548G	SD648G	SD546G		
88	SD460G	SD360G	SD462G		
89	SD540G	SD640G	SD542G		
90	SD462G	SD360G	SD460G		
91	SD542G	SD640G	SD540G		
92	SD568G	SD468G	SD546G		
93	SD568G	SD466G	SD548G		
94	SD560G	SD460G	SD542G		
95	SD560G	SD462G	SD540G		
96	SD648G	SD748G		SD629W	
97	SD640G	SD740G		SD639W	
98	SD640G	SD748G			SE219W
99	SD668G	SD568G		SD629W	
100	SD668G	SD560G			SE219W
101	SD660G	SD560G		SD639W	
102	SD748G	SE148G			
103	SD768G	SD668G			
104	SD740G	SE140G			
105	SD760G	SD660G			
106	SE148G	SE248G			
107	SE168G	SD768G			
108	SE248G	SE748G		SE219W,SE229W	
109	SE248G	SE740G		SF219W	SF739W,SD629W
110	SE140G	SE240G			
111	SE160G	SD760G			
112	SE240G	SE740G		SE209W,SE239W	
113	SE240G	SE748G	SE260G	SD629W,SD429W	SD429W
114	SE268G	SE168G		SE229W,SE219W	
115	SE268G	SE160G	SE260G	SD629W,SD429W	SD429W
116	SE748G	SE848G		SE719W,SE729W	
117	SE748G	SE840G	SE740G	SF219W	SF739W
118	SE260G	SE160G		SE239W,SE209W	
119	SE260G	SE168G		SF219W	SF739W,SD629W
120	SE740G	SE840G		SE709W,SE739W	
121	SE740G	SE848G		SD629W,SD429W	SD429W,SF219W
122	SE768G	SE268G		SE729W,SE719W	
123	SE768G	SE260G		SD629W,SD429W	SD429W,SF219W
124	SE848G	SF148G			
125	SE868G	SE768G			
126	SE760G	SE260G		SE739W,SE709W	
127	SE760G	SE268G	SE740G	SF219W	SF739W
128	SE840G	SF140G			
129	SE860G	SE760G			
130	SF148G	SF248G			
131	SF168G	SE868G			
132	SF248G	SF748G		SF219W	
133	SF248G	SF340G	SF268G		SE729W
134	SF140G	SF240G			
135	SF160G	SE860G			

136	SF240G	SF340G		SF209W		
137	SF268G	SF168G		SF219W		
138	SF260G	SF160G		SF209W		
139	SF260G	SF168G	SF268G			SE729W
140	SF340G	SF640G	SF362G			
141	SF340G	SF642G	SF360G			
142	SF360G	SF260G	SF362G			
143	SF362G	SF260G	SF360G			
144	SF660G	SF360G	SF642G			
145	SF660G	SF362G	SF640G			
146	SF640G	SF740G	SF642G			
147	SF642G	SF740G	SF640G			
148	SF748G	SF848G		SF729W		
149	SF740G	SF840G		SF739W		
150	SF740G	SF848G	SF748G	SG239W		SG719W
151	SF768G	SF268G		SF729W		
152	SF768G	SF660G	SF748G	SG239W		SG719W
153	SF760G	SF660G		SF739W		
154	SF848G	SG148G				
155	SF868G	SF768G				
156	SF840G	SG140G				
157	SF860G	SF760G				
158	SG148G	SG248G				
159	SG168G	SF868G				
160	SG248G	SG348G		SG229W		
161	SG140G	SG240G				
162	SG160G	SF860G				
163	SG240G	SG740G		SG239W		
164	SG240G	SG348G	SG260G	SF729W		SF209W
165	SG268G	SG168G		SG229W		
166	SG268G	SG160G	SG260G	SF729W		SF209W
167	SG260G	SG160G		SG239W		
168	SG348G	SG648G	SG366G			
169	SG348G	SG646G	SG368G			
170	SG366G	SG268G	SG368G			
171	SG646G	SG748G	SG648G			
172	SG368G	SG268G	SG366G			
173	SG648G	SG748G	SG646G			
174	SG668G	SG368G	SG646G			
175	SG668G	SG366G	SG648G			
176	SG748G	SG848G		SG719W		
177	SG748G	SG840G	SG740G	SH219W		
178	SG740G	SG840G		SG709W		
179	SG768G	SG668G		SG719W		
180	SG760G	SG260G		SG709W		
181	SG760G	SG668G	SG740G	SH219W		
182	SG848G	SH148G				
183	SG868G	SG768G				
184	SG840G	SH140G				
185	SG860G	SG760G				
186	SH148G	SH248G				
187	SH168G	SG868G				
188	SH248G	SH340G		SG709W		SG229W
189	SH140G	SH240G				
190	SH160G	SG860G				
191	SH240G	SH340G				SG709W
192	SH260G	SH160G				SG709W
193	SH260G	SH168G		SG709W		SG229W
194	SH360G	SH260G				

The following text file is generated by the program using conventional approach which shows the aspect information.

Line_Plan_Aspect_Information(Conventional).txt

```
This file is generated using Control Table Generator program.
Generated on Thursday, 14 October 2021 21:15:47
-----------------------------------------------------------------------------------------------------

Topology: A Line Plan
Control Table Section: Aspect Information
Algorithm Design: Conventional Approach
```

Route Number	Departure Signal	Destination Signal	GREEN Aspect Requirements	YELLOW Aspect Requirements
1	SA140G	SA340G	SA340G GREEN,SA340G YELLOW	SA340G GREEN,SA340G YELLOW,SA340G RED
2	SA340G	SA840G	SA840G GREEN	SA840G GREEN,SA840G RED
3	SA340G	SA848G		SA848G GREEN,SA848G RED
4	SA360G	SA160G	SA160G GREEN	SA160G GREEN,SA160G RED
5	SA840G	SB140G	SB140G GREEN,SB140G YELLOW	
6	SA860G	SA360G	SA360G GREEN,SA360G YELLOW	SA360G GREEN,SA360G YELLOW,SA360G RED
7	SA368G	SA160G		SA160G GREEN,SA160G RED
8	SA848G	SB148G	SB148G GREEN,SB148G YELLOW	
9	SA868G	SA368G	SA368G YELLOW,SA368G GREEN	SA368G YELLOW,SA368G RED
10	SB148G	SB248G	SB248G GREEN,SB248G YELLOW	SB248G GREEN,SB248G YELLOW,SB248G RED
11	SB168G	SA868G	SA868G GREEN,SA868G YELLOW	
12	SB248G	SB348G	SB348G GREEN,SB348G YELLOW	SB348G GREEN,SB348G YELLOW,SB348G RED
13	SB140G	SB240G	SB240G GREEN,SB240G YELLOW	SB240G GREEN,SB240G YELLOW,SB240G RED
14	SB160G	SA860G	SA860G GREEN,SA860G YELLOW	
15	SB240G	SB340G	SB340G GREEN,SB340G YELLOW	SB340G GREEN,SB340G YELLOW,SB340G RED
16	SB240G	SB348G		SB348G GREEN,SB348G YELLOW,SB348G RED
17	SB268G	SB168G	SB168G GREEN	SB168G GREEN,SB168G RED
18	SB268G	SB160G		SB160G GREEN,SB160G RED
19	SB260G	SB160G	SB160G GREEN	SB160G GREEN,SB160G RED
20	SB348G	SB648G	SB648G GREEN,SB648G YELLOW	SB648G GREEN,SB648G YELLOW,SB648G RED
21	SB348G	SB646G		SB646G YELLOW,SB646G RED
22	SB340G	SB640G	SB640G GREEN,SB640G YELLOW	SB640G GREEN,SB640G YELLOW,SB640G RED
23	SB340G	SB642G		SB642G YELLOW,SB642G RED
24	SB366G	SB268G		SB268G GREEN,SB268G YELLOW,SB268G RED
25	SB646G	SB748G		SB748G GREEN,SB748G YELLOW,SB748G RED
26	SB368G	SB268G	SB268G GREEN,SB268G YELLOW	SB268G GREEN,SB268G YELLOW,SB268G RED
27	SB648G	SB748G	SB748G GREEN,SB748G YELLOW	SB748G GREEN,SB748G YELLOW,SB748G RED
28	SB360G	SB260G	SB260G GREEN,SB260G YELLOW	SB260G GREEN,SB260G YELLOW,SB260G RED
29	SB640G	SB740G	SB740G GREEN,SB740G YELLOW	SB740G GREEN,SB740G YELLOW,SB740G RED
30	SB362G	SB260G		SB260G GREEN,SB260G YELLOW,SB260G RED
31	SB642G	SB740G		SB740G GREEN,SB740G YELLOW,SB740G RED
32	SB668G	SB368G	SB368G GREEN,SB368G YELLOW	SB368G GREEN,SB368G YELLOW,SB368G RED
33	SB668G	SB366G		SB366G YELLOW,SB366G RED
34	SB660G	SB360G	SB360G GREEN,SB360G YELLOW	SB360G GREEN,SB360G YELLOW,SB360G RED
35	SB660G	SB362G		SB362G YELLOW,SB362G RED
36	SB748G	SB848G	SB848G GREEN	SB848G GREEN,SB848G RED
37	SB748G	SB840G		SB840G GREEN,SB840G RED
38	SB740G	SB840G	SB840G GREEN	SB840G GREEN,SB840G RED
39	SB768G	SB668G	SB668G GREEN,SB668G YELLOW	SB668G GREEN,SB668G YELLOW,SB668G RED
40	SB760G	SB660G	SB660G GREEN,SB660G YELLOW	SB660G GREEN,SB660G YELLOW,SB660G RED
41	SB760G	SB668G		SB668G GREEN,SB668G YELLOW,SB668G RED
42	SB848G	SC148G	SC148G GREEN,SC148G YELLOW	
43	SB868G	SB768G	SB768G GREEN,SB768G YELLOW	SB768G GREEN,SB768G YELLOW,SB768G RED
44	SB840G	SC140G	SC140G GREEN,SC140G YELLOW	

45	SB860G	SB760G	SB760G GREEN,SB760G YELLOW	SB760G GREEN,SB760G YELLOW,SB760G RED
46	SC148G	SC448G	SC448G GREEN,SC448G YELLOW	SC448G GREEN,SC448G YELLOW,SC448G RED
47	SC168G	SB868G	SB868G GREEN,SB868G YELLOW	
48	SC448G	SC648G	SC648G GREEN,SC648G YELLOW	SC648G GREEN,SC648G YELLOW,SC648G RED
49	SC448G	SC640G		SC640G GREEN,SC640G YELLOW,SC640G RED
50	SC140G	SC440G	SC440G GREEN,SC440G YELLOW	SC440G GREEN,SC440G YELLOW,SC440G RED
51	SC160G	SB860G	SB860G GREEN,SB860G YELLOW	
52	SC440G	SC640G	SC640G GREEN,SC640G YELLOW	SC640G GREEN,SC640G YELLOW,SC640G RED
53	SC440G	SC648G		SC648G GREEN,SC648G YELLOW,SC648G RED
54	SC468G	SC168G	SC168G GREEN	SC168G GREEN,SC168G RED
55	SC468G	SC160G		SC160G GREEN,SC160G RED
56	SC460G	SC160G	SC160G GREEN	SC160G GREEN,SC160G RED
57	SC460G	SC168G		SC168G GREEN,SC168G RED
58	SC648G	SC848G	SC848G GREEN	SC848G GREEN,SC848G RED
59	SC648G	SC840G		SC840G GREEN,SC840G RED
60	SC640G	SC840G	SC840G GREEN	SC840G GREEN,SC840G RED
61	SC640G	SC848G		SC848G GREEN,SC848G RED
62	SC668G	SC468G	SC468G GREEN,SC468G YELLOW	SC468G GREEN,SC468G YELLOW,SC468G RED
63	SC668G	SC460G		SC460G GREEN,SC460G YELLOW,SC460G RED
64	SC660G	SC460G	SC460G GREEN,SC460G YELLOW	SC460G GREEN,SC460G YELLOW,SC460G RED
65	SC660G	SC468G		SC468G GREEN,SC468G YELLOW,SC468G RED
66	SC848G	SD148G	SD148G GREEN,SD148G YELLOW	
67	SC868G	SC668G	SC668G GREEN,SC668G YELLOW	SC668G GREEN,SC668G YELLOW,SC668G RED
68	SC840G	SD140G	SD140G GREEN,SD140G YELLOW	
69	SC860G	SC660G	SC660G GREEN,SC660G YELLOW	SC660G GREEN,SC660G YELLOW,SC660G RED
70	SD148G	SD348G	SD348G GREEN,SD348G YELLOW	SD348G GREEN,SD348G YELLOW,SD348G RED
71	SD168G	SC868G	SC868G GREEN,SC868G YELLOW	
72	SD348G	SD448G	SD448G GREEN,SD448G YELLOW	SD448G GREEN,SD448G YELLOW,SD448G RED
73	SD348G	SD440G		SD440G GREEN,SD440G YELLOW,SD440G RED
74	SD140G	SD340G	SD340G GREEN,SD340G YELLOW	SD340G GREEN,SD340G YELLOW,SD340G RED
75	SD160G	SC860G	SC860G GREEN,SC860G YELLOW	
76	SD340G	SD440G	SD440G GREEN,SD440G YELLOW	SD440G GREEN,SD440G YELLOW,SD440G RED
77	SD368G	SD168G	SD168G GREEN	SD168G GREEN,SD168G RED
78	SD360G	SD160G	SD160G GREEN	SD160G GREEN,SD160G RED
79	SD360G	SD168G		SD168G GREEN,SD168G RED
80	SD448G	SD548G	SD548G GREEN,SD548G YELLOW	SD548G GREEN,SD548G YELLOW,SD548G RED
81	SD448G	SD546G		SD546G YELLOW,SD546G RED
82	SD440G	SD540G	SD540G GREEN,SD540G YELLOW	SD540G GREEN,SD540G YELLOW,SD540G RED
83	SD440G	SD542G		SD542G YELLOW,SD542G RED
84	SD466G	SD368G		SD368G GREEN,SD368G YELLOW,SD368G RED
85	SD546G	SD648G		SD648G GREEN,SD648G YELLOW,SD648G RED
86	SD468G	SD368G	SD368G GREEN,SD368G YELLOW	SD368G GREEN,SD368G YELLOW,SD368G RED
87	SD548G	SD648G	SD648G GREEN,SD648G YELLOW	SD648G GREEN,SD648G YELLOW,SD648G RED
88	SD460G	SD360G	SD360G GREEN,SD360G YELLOW	SD360G GREEN,SD360G YELLOW,SD360G RED
89	SD540G	SD640G	SD640G GREEN,SD640G YELLOW	SD640G GREEN,SD640G YELLOW,SD640G RED
90	SD462G	SD360G		SD360G GREEN,SD360G YELLOW,SD360G RED
91	SD542G	SD640G		SD640G GREEN,SD640G YELLOW,SD640G RED
92	SD568G	SD468G	SD468G GREEN,SD468G YELLOW	SD468G GREEN,SD468G YELLOW,SD468G RED
93	SD568G	SD466G		SD466G YELLOW,SD466G RED
94	SD560G	SD460G	SD460G GREEN,SD460G YELLOW	SD460G GREEN,SD460G YELLOW,SD460G RED
95	SD560G	SD462G		SD462G YELLOW,SD462G RED
96	SD648G	SD748G	SD748G GREEN	SD748G GREEN,SD748G RED
97	SD640G	SD740G	SD740G GREEN	SD740G GREEN,SD740G RED
98	SD640G	SD748G		SD748G GREEN,SD748G RED
99	SD668G	SD568G	SD568G GREEN,SD568G YELLOW	SD568G GREEN,SD568G YELLOW,SD568G RED
100	SD668G	SD560G		SD560G GREEN,SD560G YELLOW,SD560G RED
101	SD660G	SD560G	SD560G GREEN,SD560G YELLOW	SD560G GREEN,SD560G YELLOW,SD560G RED
102	SD748G	SE148G	SE148G GREEN,SE148G YELLOW	
103	SD768G	SD668G	SD668G GREEN,SD668G YELLOW	SD668G GREEN,SD668G YELLOW,SD668G RED
104	SD740G	SE140G	SE140G GREEN,SE140G YELLOW	
105	SD760G	SD660G	SD660G GREEN,SD660G YELLOW	SD660G GREEN,SD660G YELLOW,SD660G RED
106	SE148G	SE248G	SE248G GREEN,SE248G YELLOW	SE248G GREEN,SE248G YELLOW,SE248G RED
107	SE168G	SD768G	SD768G GREEN,SD768G YELLOW	

108	SE248G	SE748G	SE748G GREEN,SE748G YELLOW	SE748G GREEN,SE748G YELLOW,SE748G RED
109	SE248G	SE740G		SE740G GREEN,SE740G YELLOW,SE740G RED
110	SE140G	SE240G	SE240G GREEN,SE240G YELLOW	SE240G GREEN,SE240G YELLOW,SE240G RED
111	SE160G	SD760G	SD760G GREEN,SD760G YELLOW	
112	SE240G	SE740G	SE740G GREEN,SE740G YELLOW	SE740G GREEN,SE740G YELLOW,SE740G RED
113	SE240G	SE748G		SE748G GREEN,SE748G YELLOW,SE748G RED
114	SE268G	SE168G	SE168G GREEN	SE168G GREEN,SE168G RED
115	SE268G	SE160G		SE160G GREEN,SE160G RED
116	SE748G	SE848G	SE848G GREEN	SE848G GREEN,SE848G RED
117	SE748G	SE840G		SE840G GREEN,SE840G RED
118	SE260G	SE160G	SE160G GREEN	SE160G GREEN,SE160G RED
119	SE260G	SE168G		SE168G GREEN,SE168G RED
120	SE740G	SE840G	SE840G GREEN	SE840G GREEN,SE840G RED
121	SE740G	SE848G		SE848G GREEN,SE848G RED
122	SE768G	SE268G	SE268G GREEN,SE268G YELLOW	SE268G GREEN,SE268G YELLOW,SE268G RED
123	SE768G	SE260G		SE260G GREEN,SE260G YELLOW,SE260G RED
124	SE848G	SF148G	SF148G GREEN,SF148G YELLOW	
125	SE868G	SE768G	SE768G GREEN,SE768G YELLOW	SE768G GREEN,SE768G YELLOW,SE768G RED
126	SE760G	SE260G	SE260G GREEN,SE260G YELLOW	SE260G GREEN,SE260G YELLOW,SE260G RED
127	SE760G	SE268G		SE268G GREEN,SE268G YELLOW,SE268G RED
128	SE840G	SF140G	SF140G GREEN,SF140G YELLOW	
129	SE860G	SE760G	SE760G GREEN,SE760G YELLOW	SE760G GREEN,SE760G YELLOW,SE760G RED
130	SF148G	SF248G	SF248G GREEN,SF248G YELLOW	SF248G GREEN,SF248G YELLOW,SF248G RED
131	SF168G	SE868G	SE868G GREEN,SE868G YELLOW	
132	SF248G	SF748G	SF748G GREEN,SF748G YELLOW	SF748G GREEN,SF748G YELLOW,SF748G RED
133	SF248G	SF340G		SF340G GREEN,SF340G YELLOW,SF340G RED
134	SF140G	SF240G	SF240G GREEN,SF240G YELLOW	SF240G GREEN,SF240G YELLOW,SF240G RED
135	SF160G	SE860G	SE860G GREEN,SE860G YELLOW	
136	SF240G	SF340G	SF340G GREEN,SF340G YELLOW	SF340G GREEN,SF340G YELLOW,SF340G RED
137	SF268G	SF168G	SF168G GREEN	SF168G GREEN,SF168G RED
138	SF260G	SF160G	SF160G GREEN	SF160G GREEN,SF160G RED
139	SF260G	SF168G		SF168G GREEN,SF168G RED
140	SF340G	SF640G	SF640G GREEN,SF640G YELLOW	SF640G GREEN,SF640G YELLOW,SF640G RED
141	SF340G	SF642G		SF642G YELLOW,SF642G RED
142	SF360G	SF260G	SF260G GREEN,SF260G YELLOW	SF260G GREEN,SF260G YELLOW,SF260G RED
143	SF362G	SF260G		SF260G GREEN,SF260G YELLOW,SF260G RED
144	SF660G	SF360G	SF360G GREEN,SF360G YELLOW	SF360G GREEN,SF360G YELLOW,SF360G RED
145	SF660G	SF362G		SF362G YELLOW,SF362G RED
146	SF640G	SF740G	SF740G GREEN,SF740G YELLOW	SF740G GREEN,SF740G YELLOW,SF740G RED
147	SF642G	SF740G		SF740G GREEN,SF740G YELLOW,SF740G RED
148	SF748G	SF848G	SF848G GREEN	SF848G GREEN,SF848G RED
149	SF740G	SF840G	SF840G GREEN	SF840G GREEN,SF840G RED
150	SF740G	SF848G		SF848G GREEN,SF848G RED
151	SF768G	SF268G	SF268G GREEN,SF268G YELLOW	SF268G GREEN,SF268G YELLOW,SF268G RED
152	SF768G	SF660G		SF660G GREEN,SF660G YELLOW,SF660G RED
153	SF760G	SF660G	SF660G GREEN,SF660G YELLOW	SF660G GREEN,SF660G YELLOW,SF660G RED
154	SF848G	SG148G	SG148G GREEN,SG148G YELLOW	
155	SF868G	SF768G	SF768G GREEN,SF768G YELLOW	SF768G GREEN,SF768G YELLOW,SF768G RED
156	SF840G	SG140G	SG140G GREEN,SG140G YELLOW	
157	SF860G	SF760G	SF760G GREEN,SF760G YELLOW	SF760G GREEN,SF760G YELLOW,SF760G RED
158	SG148G	SG248G	SG248G GREEN,SG248G YELLOW	SG248G GREEN,SG248G YELLOW,SG248G RED
159	SG168G	SF868G	SF868G GREEN,SF868G YELLOW	
160	SG248G	SG348G	SG348G GREEN,SG348G YELLOW	SG348G GREEN,SG348G YELLOW,SG348G RED
161	SG140G	SG240G	SG240G GREEN,SG240G YELLOW	SG240G GREEN,SG240G YELLOW,SG240G RED
162	SG160G	SF860G	SF860G GREEN,SF860G YELLOW	
163	SG240G	SG740G	SG740G GREEN,SG740G YELLOW	SG740G GREEN,SG740G YELLOW,SG740G RED
164	SG240G	SG348G		SG348G GREEN,SG348G YELLOW,SG348G RED
165	SG268G	SG168G	SG168G GREEN	SG168G GREEN,SG168G RED
166	SG268G	SG160G		SG160G GREEN,SG160G RED
167	SG260G	SG160G	SG160G GREEN	SG160G GREEN,SG160G RED
168	SG348G	SG648G	SG648G GREEN,SG648G YELLOW	SG648G GREEN,SG648G YELLOW,SG648G RED
169	SG348G	SG646G		SG646G YELLOW,SG646G RED
170	SG366G	SG268G		SG268G GREEN,SG268G YELLOW,SG268G RED

171	SG646G	SG748G		SG748G GREEN,SG748G YELLOW,SG748G RED
172	SG368G	SG268G	SG268G GREEN,SG268G YELLOW	SG268G GREEN,SG268G YELLOW,SG268G RED
173	SG648G	SG748G	SG748G GREEN,SG748G YELLOW	SG748G GREEN,SG748G YELLOW,SG748G RED
174	SG668G	SG368G	SG368G GREEN,SG368G YELLOW	SG368G GREEN,SG368G YELLOW,SG368G RED
175	SG668G	SG366G		SG366G YELLOW,SG366G RED
176	SG748G	SG848G	SG848G GREEN	SG848G GREEN,SG848G RED
177	SG748G	SG840G		SG840G GREEN,SG840G RED
178	SG740G	SG840G	SG840G GREEN	SG840G GREEN,SG840G RED
179	SG768G	SG668G	SG668G GREEN,SG668G YELLOW	SG668G GREEN,SG668G YELLOW,SG668G RED
180	SG760G	SG260G	SG260G GREEN,SG260G YELLOW	SG260G GREEN,SG260G YELLOW,SG260G RED
181	SG760G	SG668G		SG668G GREEN,SG668G YELLOW,SG668G RED
182	SG848G	SH148G	SH148G GREEN,SH148G YELLOW	
183	SG868G	SG768G	SG768G GREEN,SG768G YELLOW	SG768G GREEN,SG768G YELLOW,SG768G RED
184	SG840G	SH140G	SH140G GREEN,SH140G YELLOW	
185	SG860G	SG760G	SG760G GREEN,SG760G YELLOW	SG760G GREEN,SG760G YELLOW,SG760G RED
186	SH148G	SH248G	SH248G YELLOW,SH248G GREEN	SH248G YELLOW,SH248G RED
187	SH168G	SG868G	SG868G GREEN,SG868G YELLOW	
188	SH248G	SH340G		SH340G GREEN,SH340G RED
189	SH140G	SH240G	SH240G GREEN,SH240G YELLOW	SH240G GREEN,SH240G YELLOW,SH240G RED
190	SH160G	SG860G	SG860G GREEN,SG860G YELLOW	
191	SH240G	SH340G	SH340G GREEN	SH340G GREEN,SH340G RED
192	SH260G	SH160G	SH160G GREEN	SH160G GREEN,SH160G RED
193	SH260G	SH168G		SH168G GREEN,SH168G RED
194	SH360G	SH260G	SH260G GREEN,SH260G YELLOW	SH260G GREEN,SH260G YELLOW,SH260G RED

A.9.3 Layout Diagram with Compositional Cut-line

This topology in Figure A.17 to A.20 show the complex station layout with the with cut-line.

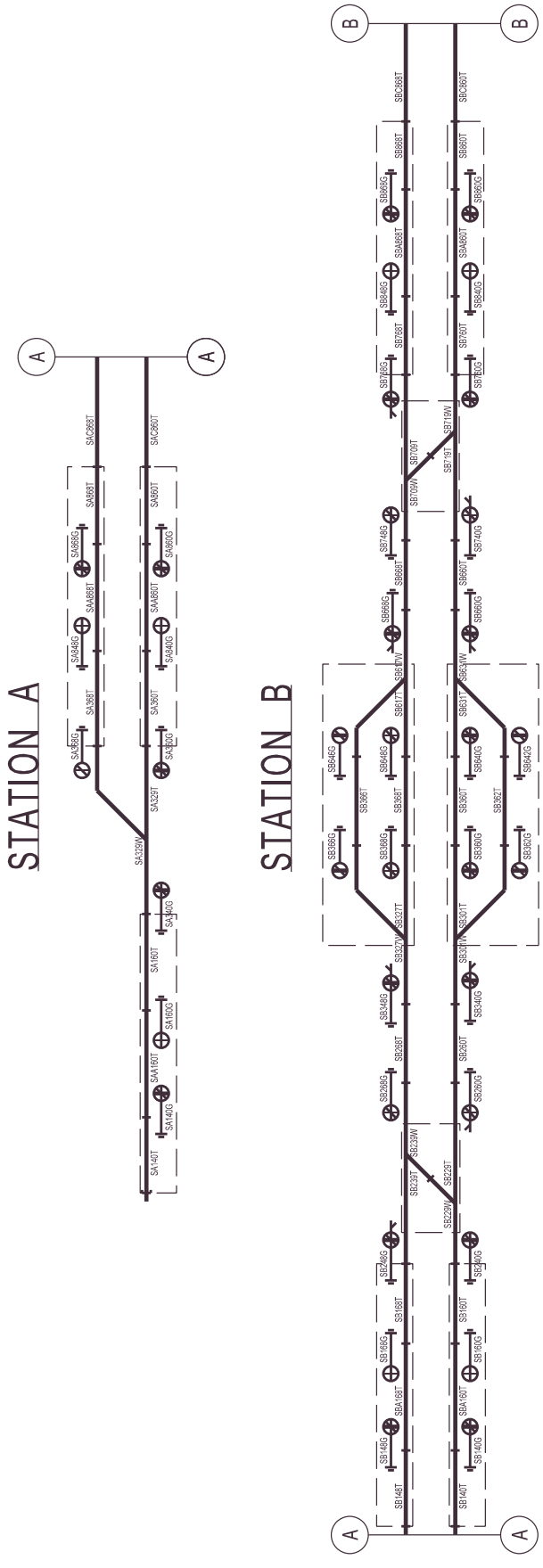


Figure A.17: The line plan with compositional cut-line (sheet 1)

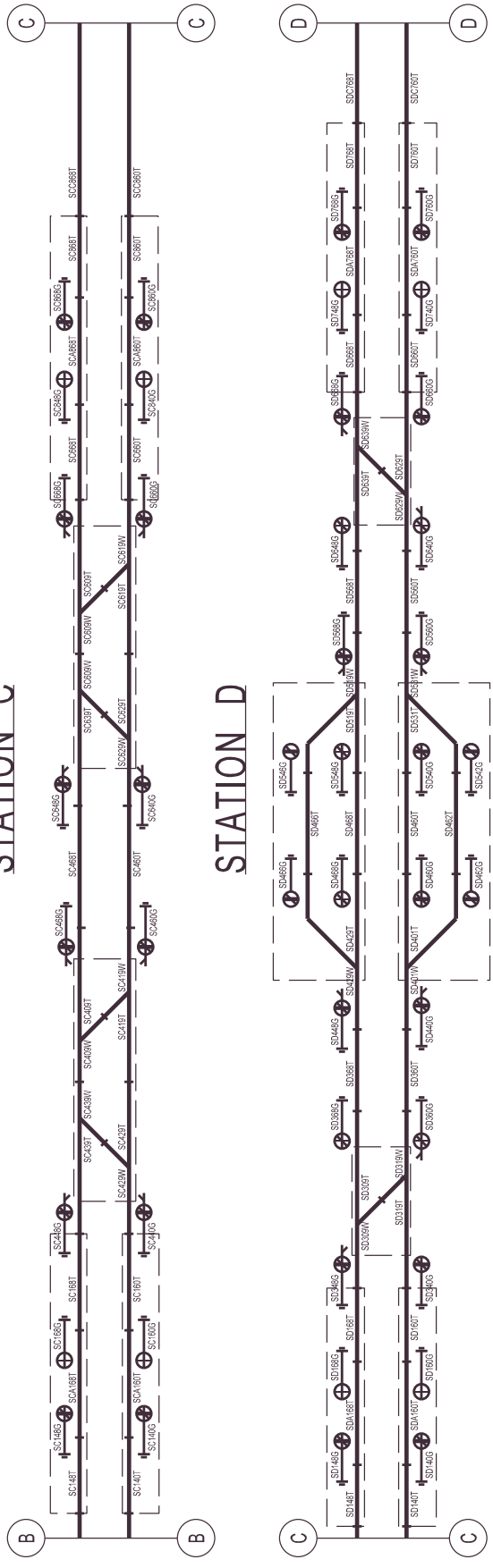


Figure A.18: The line plan with compositional cut-line (sheet 2)

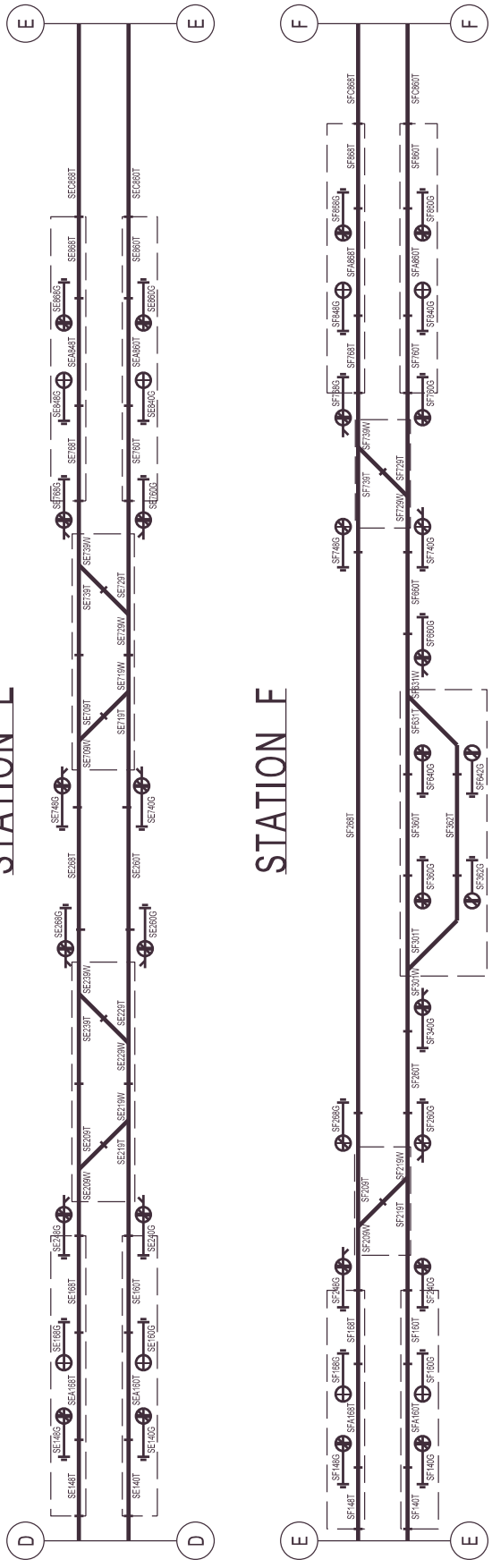


Figure A.19: The line plan with compositional cut-line (sheet 3)

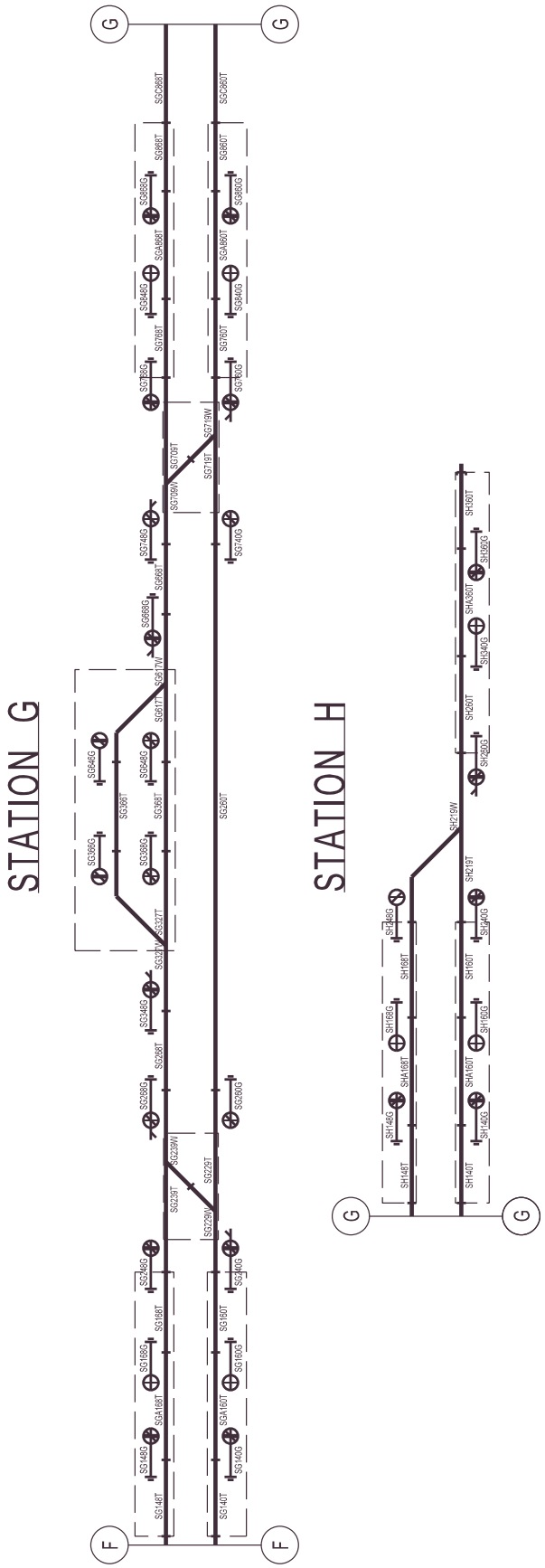


Figure A.20: The line plan with compositional cut-line (sheet 4)

A.9.4 Output Files using Compositional Approach

The following text file is generated by the program using compositional approach which shows the route information.

Line_Plan_Route_Information(Compositional).txt

```
This file is generated using Control Table Generator program.
Generated on Friday, 15 October 2021 20:48:36
-----------------------------------------------------------------------------------------------------

Topology: A Line Plan
Control Table Section: Route Information
Algorithm Design: Compositional Approach
```

Route Number	Departure Signal	Destination Signal	Tracks Required	Points Required In Straight Position	Points Required In Turnout Position
1	SA140G	SA340G	SAA160T,SA160T		
2	SA340G	SA840G	SA329T,SA360T	SA329W	
3	SA340G	SA848G	SA329T,SA368T		SA329W
4	SA840G	SB140G	SAA860T,SA860T,SAC860T,SB140T		
5	SA848G	SB148G	SAA868T,SA868T,SAC868T,SB148T		
6	SA868G	SA368G	SAA868T,SA368T		
7	SA368G	SA160G	SA329T,SA160T		SA329W
8	SA860G	SA360G	SAA860T,SA360T		
9	SA360G	SA160G	SA329T,SA160T	SA329W	
10	SB148G	SB248G	SBA168T,SB168T		
11	SB248G	SB348G	SB239T,SB268T	SB239W	
12	SB140G	SB240G	SBA160T,SB160T		
13	SB240G	SB340G	SB229T,SB260T	SB229W	
14	SB240G	SB348G	SB229T,SB239T,SB268T		SB229W,SB239W
15	SB868G	SB768G	SBA868T,SB768T		
16	SB768G	SB668G	SB709T,SB668T	SB709W	
17	SB860G	SB760G	SBA860T,SB760T		
18	SB760G	SB660G	SB719T,SB660T	SB719W	
19	SB760G	SB668G	SB719T,SB709T,SB668T		SB719W,SB709W
20	SC148G	SC448G	SCA168T,SC168T		
21	SC448G	SC648G	SC439T,SC409T,SC468T	SC439W,SC409W	
22	SC448G	SC640G	SC439T,SC409T,SC419T,SC460T	SC439W	SC409W,SC419W
23	SC140G	SC440G	SCA160T,SC160T		
24	SC440G	SC640G	SC429T,SC419T,SC460T	SC429W,SC419W	
25	SC440G	SC648G	SC429T,SC439T,SC409T,SC468T	SC409W	SC429W,SC439W
26	SC868G	SC668G	SCA868T,SC668T		
27	SC668G	SC468G	SC609T,SC639T,SC468T	SC609W,SC639W	
28	SC668G	SC460G	SC609T,SC639T,SC629T,SC460T	SC609W	SC639W,SC629W
29	SC860G	SC660G	SCA860T,SC660T		
30	SC660G	SC460G	SC619T,SC629T,SC460T	SC619W,SC629W	
31	SC660G	SC468G	SC619T,SC609T,SC639T,SC468T	SC639W	SC619W,SC609W
32	SD148G	SD348G	SDA168T,SD168T		
33	SD348G	SD448G	SD309T,SD368T	SD309W	
34	SD348G	SD440G	SD309T,SD319T,SD360T		SD309W,SD319W
35	SD140G	SD340G	SDA160T,SD160T		
36	SD340G	SD440G	SD319T,SD360T	SD319W	
37	SD768G	SD668G	SDA768T,SD668T		
38	SD668G	SD568G	SD639T,SD568T	SD639W	
39	SD668G	SD560G	SD639T,SD629T,SD560T		SD639W,SD629W
40	SD760G	SD660G	SDA760T,SD660T		

41	SD660G	SD560G	SD629T,SD560T	SD629W	
42	SE148G	SE248G	SEA168T,SE168T		
43	SE248G	SE748G	SE209T,SE239T,SE268T	SE209W,SE239W	
44	SE248G	SE740G	SE209T,SE219T,SE229T,SE260T	SE229W	SE209W,SE219W
45	SE140G	SE240G	SEA160T,SE160T		
46	SE240G	SE740G	SE219T,SE229T,SE260T	SE219W,SE229W	
47	SE240G	SE748G	SE219T,SE229T,SE239T,SE268T	SE219W	SE229W,SE239W
48	SE868G	SE768G	SEA868T,SE768T		
49	SE768G	SE268G	SE739T,SE709T,SE268T	SE739W,SE709W	
50	SE768G	SE260G	SE739T,SE729T,SE719T,SE260T	SE719W	SE739W,SE729W
51	SE860G	SE760G	SEA860T,SE760T		
52	SE760G	SE260G	SE729T,SE719T,SE260T	SE729W,SE719W	
53	SE760G	SE268G	SE729T,SE719T,SE709T,SE268T	SE729W	SE719W,SE709W
54	SF148G	SF248G	SFA168T,SF168T		
55	SF248G	SF748G	SF209T,SF268T	SF209W	
56	SF248G	SF340G	SF209T,SF219T,SF260T		SF209W,SF219W
57	SF140G	SF240G	SFA160T,SF160T		
58	SF240G	SF340G	SF219T,SF260T	SF219W	
59	SF868G	SF768G	SFA868T,SF768T		
60	SF768G	SF268G	SF739T,SF268T	SF739W	
61	SF768G	SF660G	SF739T,SF729T,SF660T		SF739W,SF729W
62	SF860G	SF760G	SFA860T,SF760T		
63	SF760G	SF660G	SF729T,SF660T	SF729W	
64	SG148G	SG248G	SGA168T,SG168T		
65	SG248G	SG348G	SG239T,SG268T	SG239W	
66	SG140G	SG240G	SGA160T,SG160T		
67	SG240G	SG740G	SG229T,SG260T	SG229W	
68	SG240G	SG348G	SG229T,SG239T,SG268T		SG229W,SG239W
69	SG868G	SG768G	SGA868T,SG768T		
70	SG768G	SG668G	SG709T,SG668T	SG709W	
71	SG860G	SG760G	SGA860T,SG760T		
72	SG760G	SG260G	SG719T,SG260T	SG719W	
73	SG760G	SG668G	SG719T,SG709T,SG668T		SG719W,SG709W
74	SH148G	SH248G	SHA168T,SH168T		
75	SH248G	SH340G	SA219T,SH260T		SH219W
76	SH140G	SH240G	SHA160T,SH160T		
77	SH240G	SH340G	SA219T,SH260T	SH219W	
78	SH360G	SH260G	SHA360T,SH260T		
79	SH260G	SH160G	SA219T,SH160T	SH219W	
80	SH260G	SH168G	SA219T,SH168T		SH219W
81	SH160G	SG860G	SHA160T,SH140T,SGC860T,SG860T		
82	SH168G	SG868G	SHA168T,SH148T,SGC868T,SG868T		
83	SB268G	SB168G	SB239T,SB168T	SB239W	
84	SB268G	SB160G	SB239T,SB229T,SB160T		SB239W,SB229W
85	SB168G	SA868G	SBA168T,SB148T,SAC868T,SA868T		
86	SB160G	SA860G	SBA160T,SB140T,SAC860T,SA860T		
87	SB348G	SB648G	SB327T,SB368T	SB327W	
88	SB348G	SB646G	SB327T,SB366T		SB327W
89	SB648G	SB748G	SB617T,SB668T	SB617W	
90	SB646G	SB748G	SB617T,SB668T		SB617W
91	SB260G	SB160G	SB229T,SB160T	SB229W	
92	SB340G	SB640G	SB301T,SB360T	SB301W	
93	SB340G	SB642G	SB301T,SB362T		SB301W
94	SB640G	SB740G	SB631T,SB660T	SB631W	
95	SB642G	SB740G	SB631T,SB660T		SB631W
96	SB668G	SB368G	SB617T,SB368T	SB617W	
97	SB668G	SB366G	SB617T,SB366T		SB617W
98	SB368G	SB268G	SB327T,SB268T	SB327W	
99	SB366G	SB268G	SB327T,SB268T		SB327W
100	SB748G	SB848G	SB709T,SB768T	SB709W	
101	SB748G	SB840G	SB709T,SB719T,SB760T		SB709W,SB719W
102	SB848G	SC148G	SBA868T,SB868T,SBC868T,SC148T		
103	SB840G	SC140G	SBA860T,SB860T,SBC860T,SC140T		

104	SB660G	SB360G	SB631T,SB360T	SB631W	
105	SB660G	SB362G	SB631T,SB362T		SB631W
106	SB360G	SB260G	SB301T,SB260T	SB301W	
107	SB362G	SB260G	SB301T,SB260T		SB301W
108	SB740G	SB840G	SB719T,SB760T	SB719W	
109	SC468G	SC168G	SC409T,SC439T,SC168T	SC409W,SC439W	
110	SC468G	SC160G	SC409T,SC439T,SC429T,SC160T	SC409W	SC439W,SC429W
111	SC168G	SB868G	SCA168T,SC148T,SBC868T,SB868T		
112	SC160G	SB860G	SCA160T,SC140T,SBC860T,SB860T		
113	SC648G	SC848G	SC639T,SC609T,SC668T	SC639W,SC609W	
114	SC648G	SC840G	SC639T,SC609T,SC619T,SC660T	SC639W	SC609W,SC619W
115	SC848G	SD148G	SCA868T,SC868T,SBC868T,SD148T		
116	SC840G	SD140G	SCA860T,SC860T,SBC860T,SD140T		
117	SC460G	SC160G	SC419T,SC429T,SC160T	SC419W,SC429W	
118	SC460G	SC168G	SC419T,SC409T,SC439T,SC168T	SC439W	SC419W,SC409W
119	SC640G	SC840G	SC629T,SC619T,SC660T	SC629W,SC619W	
120	SC640G	SC848G	SC629T,SC639T,SC609T,SC668T	SC609W	SC629W,SC639W
121	SD368G	SD168G	SD309T,SD168T	SD309W	
122	SD168G	SC868G	SDA168T,SD148T,SBC868T,SC868T		
123	SD448G	SD548G	SD429T,SD468T	SD429W	
124	SD448G	SD546G	SD429T,SD466T		SD429W
125	SD548G	SD648G	SD519T,SD568T	SD519W	
126	SD546G	SD648G	SD519T,SD568T		SD519W
127	SD360G	SD160G	SD319T,SD160T	SD319W	
128	SD360G	SD168G	SD319T,SD309T,SD168T		SD319W,SD309W
129	SD160G	SC860G	SDA160T,SD140T,SBC860T,SC860T		
130	SD440G	SD540G	SD401T,SD460T	SD401W	
131	SD440G	SD542G	SD401T,SD462T		SD401W
132	SD540G	SD640G	SD531T,SD560T	SD531W	
133	SD542G	SD640G	SD531T,SD560T		SD531W
134	SD568G	SD468G	SD519T,SD468T	SD519W	
135	SD568G	SD466G	SD519T,SD466T		SD519W
136	SD468G	SD368G	SD429T,SD368T	SD429W	
137	SD466G	SD368G	SD429T,SD368T		SD429W
138	SD648G	SD748G	SD639T,SD668T	SD639W	
139	SD748G	SE148G	SDA768T,SD768T,SDC768T,SE148T		
140	SD560G	SD460G	SD531T,SD460T	SD531W	
141	SD560G	SD462G	SD531T,SD462T		SD531W
142	SD460G	SD360G	SD401T,SD360T	SD401W	
143	SD462G	SD360G	SD401T,SD360T		SD401W
144	SD640G	SD740G	SD629T,SD660T	SD629W	
145	SD640G	SD748G	SD629T,SD639T,SD668T		SD629W,SD639W
146	SD740G	SE140G	SDA760T,SD760T,SDC760T,SE140T		
147	SE268G	SE168G	SE239T,SE209T,SE168T	SE239W,SE209W	
148	SE268G	SE160G	SE239T,SE229T,SE219T,SE160T	SE219W	SE239W,SE229W
149	SE168G	SD768G	SEA168T,SE148T,SDC768T,SD768T		
150	SE160G	SD760G	SEA160T,SE140T,SDC760T,SD760T		
151	SE748G	SE848G	SE709T,SE739T,SE768T	SE709W,SE739W	
152	SE748G	SE840G	SE709T,SE719T,SE729T,SE760T	SE729W	SE709W,SE719W
153	SE848G	SF148G	SEA868T,SE868T,SEC868T,SF148T		
154	SE840G	SF140G	SEA860T,SE860T,SEC860T,SF140T		
155	SE260G	SE160G	SE229T,SE219T,SE160T	SE229W,SE219W	
156	SE260G	SE168G	SE229T,SE219T,SE209T,SE168T	SE229W	SE219W,SE209W
157	SE740G	SE840G	SE719T,SE729T,SE760T	SE719W,SE729W	
158	SE740G	SE848G	SE719T,SE729T,SE739T,SE768T	SE719W	SE729W,SE739W
159	SF268G	SF168G	SF209T,SF168T	SF209W	
160	SF168G	SE868G	SFA168T,SF148T,SEC868T,SE868T		
161	SF748G	SF848G	SF739T,SF768T	SF739W	
162	SF848G	SG148G	SFA868T,SF868T,SFC868T,SG148T		
163	SF260G	SF160G	SF219T,SF160T	SF219W	
164	SF260G	SF168G	SF219T,SF209T,SF168T		SF219W,SF209W
165	SF160G	SE860G	SFA160T,SF140T,SEC860T,SE860T		
166	SF340G	SF640G	SF301T,SF360T	SF301W	

167	SF340G	SF642G	SF301T,SF362T		SF301W
168	SF640G	SF740G	SF631T,SF660T	SF631W	
169	SF642G	SF740G	SF631T,SF660T		SF631W
170	SF660G	SF360G	SF631T,SF360T	SF631W	
171	SF660G	SF362G	SF631T,SF362T		SF631W
172	SF360G	SF260G	SF301T,SF260T	SF301W	
173	SF362G	SF260G	SF301T,SF260T		SF301W
174	SF740G	SF840G	SF729T,SF760T	SF729W	
175	SF740G	SF848G	SF729T,SF739T,SF768T		SF729W,SF739W
176	SF840G	SG140G	SFA860T,SF860T,SFC860T,SG140T		
177	SG268G	SG168G	SG239T,SG168T	SG239W	
178	SG268G	SG160G	SG239T,SG229T,SG160T		SG239W,SG229W
179	SG168G	SF868G	SGA168T,SG148T,SFC868T,SF868T		
180	SG160G	SF860G	SGA160T,SG140T,SFC860T,SF860T		
181	SG348G	SG648G	SG327T,SG368T	SG327W	
182	SG348G	SG646G	SG327T,SG366T		SG327W
183	SG648G	SG748G	SG617T,SG668T	SG617W	
184	SG646G	SG748G	SG617T,SG668T		SG617W
185	SG260G	SG160G	SG229T,SG160T	SG229W	
186	SG668G	SG368G	SG617T,SG368T	SG617W	
187	SG668G	SG366G	SG617T,SG366T		SG617W
188	SG368G	SG268G	SG327T,SG268T	SG327W	
189	SG366G	SG268G	SG327T,SG268T		SG327W
190	SG748G	SG848G	SG709T,SG768T	SG709W	
191	SG748G	SG840G	SG709T,SG719T,SG760T		SG709W,SG719W
192	SG848G	SH148G	SGA868T,SG868T,SGC868T,SH148T		
193	SG840G	SH140G	SGA860T,SG860T,SGC860T,SH140T		
194	SG740G	SG840G	SG719T,SG760T	SG719W	

The following text file is generated by the program using compositional approach which shows the flank protection information.

Line_Plan_Flank_Information(Compositional).txt

```
This file is generated using Control Table Generator program.
Generated on Friday, 15 October 2021 20:48:36
-----------------------------------------------------------------------------------------------------

Topology: A Line Plan
Control Table Section: Flank Protection Information
Algorithm Design: Compositional Approach

Route      Departure    Destination   Flank by               Flank by Points         Flank by Points
Number     Signal       Signal        Signals                In Straight Position    In Turnout Position
-----------------------------------------------------------------------------------------------------
1          SA140G       SA340G
2          SA340G       SA840G                                                       SB239W
3          SA340G       SA848G        SA360G                 SC439W                  SC439W,SC419W
4          SA840G       SB140G
5          SA848G       SB148G
6          SA868G       SA368G
7          SA368G       SA160G        SA360G                 SC439W                  SC439W,SC419W
8          SA860G       SA360G
9          SA360G       SA160G                                                       SB239W
```

10	SB148G	SB248G			
11	SB248G	SB348G		SB229W	
12	SB140G	SB240G			
13	SB240G	SB340G		SB239W	
14	SB240G	SB348G	SB260G	SA329W	
15	SB868G	SB768G			
16	SB768G	SB668G		SB719W	
17	SB860G	SB760G			
18	SB760G	SB660G		SB709W	
19	SB760G	SB668G	SB740G		SC439W
20	SC148G	SC448G			
21	SC448G	SC648G		SC429W,SC419W	
22	SC448G	SC640G	SC468G,SC440G	SA329W	
23	SC140G	SC440G			
24	SC440G	SC640G		SC439W,SC409W	
25	SC440G	SC648G	SC460G		SB709W
26	SC868G	SC668G			
27	SC668G	SC468G		SC619W,SC629W	
28	SC668G	SC460G	SC648G		SD319W
29	SC860G	SC660G			
30	SC660G	SC460G		SC609W,SC639W	
31	SC660G	SC468G	SC640G,SC668G	SE219W	SE219W,SE239W
32	SD148G	SD348G			
33	SD348G	SD448G		SD319W	
34	SD348G	SD440G	SD368G	SC609W	SC629W
35	SD140G	SD340G			
36	SD340G	SD440G		SD309W	
37	SD768G	SD668G			
38	SD668G	SD568G		SD629W	
39	SD668G	SD560G	SD648G		SE219W
40	SD760G	SD660G			
41	SD660G	SD560G		SD639W	
42	SE148G	SE248G			
43	SE248G	SE748G		SE219W,SE229W	
44	SE248G	SE740G	SE268G		SD629W
45	SE140G	SE240G			
46	SE240G	SE740G		SE209W,SE239W	
47	SE240G	SE748G	SE260G,SE248G	SC609W	SC629W,SC609W
48	SE868G	SE768G			
49	SE768G	SE268G		SE729W,SE719W	
50	SE768G	SE260G	SE748G		SF219W
51	SE860G	SE760G			
52	SE760G	SE260G		SE739W,SE709W	
53	SE760G	SE268G	SE740G,SE768G	SH219W	SG239W
54	SF148G	SF248G			
55	SF248G	SF748G		SF219W	
56	SF248G	SF340G	SF268G		SE729W
57	SF140G	SF240G			
58	SF240G	SF340G		SF209W	
59	SF868G	SF768G			
60	SF768G	SF268G		SF729W	
61	SF768G	SF660G	SF748G,SF760G	SH219W	
62	SF860G	SF760G			
63	SF760G	SF660G		SF739W	
64	SG148G	SG248G			
65	SG248G	SG348G		SG229W	
66	SG140G	SG240G			
67	SG240G	SG740G		SG239W	
68	SG240G	SG348G	SG260G,SG248G	SE729W	SE729W,SE709W
69	SG868G	SG768G			
70	SG768G	SG668G		SG719W	
71	SG860G	SG760G			
72	SG760G	SG260G		SG709W	

73	SG760G	SG668G	SG740G	SH219W	
74	SH148G	SH248G			
75	SH248G	SH340G	SH240G	SE729W	SE729W,SE709W,SF729W
76	SH140G	SH240G			
77	SH240G	SH340G			SG709W
78	SH360G	SH260G			
79	SH260G	SH160G			SG709W
80	SH260G	SH168G	SH240G	SE729W	SE729W,SE709W,SF729W
81	SH160G	SG860G			
82	SH168G	SG868G			
83	SB268G	SB168G		SB229W	
84	SB268G	SB160G	SB260G	SA329W	
85	SB168G	SA868G			
86	SB160G	SA860G			
87	SB348G	SB648G	SB366G		
88	SB348G	SB646G	SB368G		
89	SB648G	SB748G	SB646G		
90	SB646G	SB748G	SB648G		
91	SB260G	SB160G		SB239W	
92	SB340G	SB640G	SB362G		
93	SB340G	SB642G	SB360G		
94	SB640G	SB740G	SB642G		
95	SB642G	SB740G	SB640G		
96	SB668G	SB368G	SB646G		
97	SB668G	SB366G	SB648G		
98	SB368G	SB268G	SB366G		
99	SB366G	SB268G	SB368G		
100	SB748G	SB848G		SB719W	
101	SB748G	SB840G	SB740G		SC439W
102	SB848G	SC148G			
103	SB840G	SC140G			
104	SB660G	SB360G	SB642G		
105	SB660G	SB362G	SB640G		
106	SB360G	SB260G	SB362G		
107	SB362G	SB260G	SB360G		
108	SB740G	SB840G		SB709W	
109	SC468G	SC168G		SC419W,SC429W	
110	SC468G	SC160G	SC460G		SB709W
111	SC168G	SB868G			
112	SC160G	SB860G			
113	SC648G	SC848G		SC629W,SC619W	
114	SC648G	SC840G	SC668G,SC640G	SE219W	SE219W,SE239W
115	SC848G	SD148G			
116	SC840G	SD140G			
117	SC460G	SC160G		SC409W,SC439W	
118	SC460G	SC168G	SC440G,SC468G	SA329W	
119	SC640G	SC840G		SC639W,SC609W	
120	SC640G	SC848G	SC648G		SD319W
121	SD368G	SD168G		SD319W	
122	SD168G	SC868G			
123	SD448G	SD548G	SD466G		
124	SD448G	SD546G	SD468G		
125	SD548G	SD648G	SD546G		
126	SD546G	SD648G	SD548G		
127	SD360G	SD160G		SD309W	
128	SD360G	SD168G	SD368G	SC609W	SC629W
129	SD160G	SC860G			
130	SD440G	SD540G	SD462G		
131	SD440G	SD542G	SD460G		
132	SD540G	SD640G	SD542G		
133	SD542G	SD640G	SD540G		
134	SD568G	SD468G	SD546G		
135	SD568G	SD466G	SD548G		

136	SD468G	SD368G	SD466G			
137	SD466G	SD368G	SD468G			
138	SD648G	SD748G		SD629W		
139	SD748G	SE148G				
140	SD560G	SD460G	SD542G			
141	SD560G	SD462G	SD540G			
142	SD460G	SD360G	SD462G			
143	SD462G	SD360G	SD460G			
144	SD640G	SD740G		SD639W		
145	SD640G	SD748G	SD648G		SE219W	
146	SD740G	SE140G				
147	SE268G	SE168G		SE229W,SE219W		
148	SE268G	SE160G	SE248G,SE260G	SC609W	SC629W,SC609W	
149	SE168G	SD768G				
150	SE160G	SD760G				
151	SE748G	SE848G		SE719W,SE729W		
152	SE748G	SE840G	SE768G,SE740G	SH219W	SG239W	
153	SE848G	SF148G				
154	SE840G	SF140G				
155	SE260G	SE160G		SE239W,SE209W		
156	SE260G	SE168G	SE268G		SD629W	
157	SE740G	SE840G		SE709W,SE739W		
158	SE740G	SE848G	SE748G		SF219W	
159	SF268G	SF168G		SF219W		
160	SF168G	SE868G				
161	SF748G	SF848G		SF729W		
162	SF848G	SG148G				
163	SF260G	SF160G		SF209W		
164	SF260G	SF168G	SF268G		SE729W	
165	SF160G	SE860G				
166	SF340G	SF640G	SF362G			
167	SF340G	SF642G	SF360G			
168	SF640G	SF740G	SF642G			
169	SF642G	SF740G	SF640G			
170	SF660G	SF360G	SF642G			
171	SF660G	SF362G	SF640G			
172	SF360G	SF260G	SF362G			
173	SF362G	SF260G	SF360G			
174	SF740G	SF840G		SF739W		
175	SF740G	SF848G	SF760G,SF748G	SH219W		
176	SF840G	SG140G				
177	SG268G	SG168G		SG229W		
178	SG268G	SG160G	SG248G,SG260G	SE729W	SE729W,SE709W	
179	SG168G	SF868G				
180	SG160G	SF860G				
181	SG348G	SG648G	SG366G			
182	SG348G	SG646G	SG368G			
183	SG648G	SG748G	SG646G			
184	SG646G	SG748G	SG648G			
185	SG260G	SG160G		SG239W		
186	SG668G	SG368G	SG646G			
187	SG668G	SG366G	SG648G			
188	SG368G	SG268G	SG366G			
189	SG366G	SG268G	SG368G			
190	SG748G	SG848G		SG719W		
191	SG748G	SG840G	SG740G	SH219W		
192	SG848G	SH148G				
193	SG840G	SH140G				
194	SG740G	SG840G		SG709W		

The following text file is generated by the program using compositional approach which shows the aspect information.

Line_Plan_Aspect_Information(Compositional).txt

```
This file is generated using Control Table Generator program.
Generated on Friday, 15 October 2021 20:48:36
-----------------------------------------------------------------------------------------------------

Topology: A Line Plan
Control Table Section: Aspect Information
Algorithm Design: Compositional Approach
```

Route Number	Departure Signal	Destination Signal	GREEN Aspect Requirements	YELLOW Aspect Requirements
1	SA140G	SA340G	SA340G GREEN,SA340G YELLOW	SA340G GREEN,SA340G YELLOW,SA340G RED
2	SA340G	SA840G	SA840G GREEN	SA840G GREEN,SA840G RED
3	SA340G	SA848G		SA848G GREEN,SA848G RED
4	SA840G	SB140G	SB140G GREEN,SB140G YELLOW	
5	SA848G	SB148G	SB148G GREEN,SB148G YELLOW	
6	SA868G	SA368G	SA368G YELLOW,SA368G GREEN	SA368G YELLOW,SA368G RED
7	SA368G	SA160G		SA160G GREEN,SA160G RED
8	SA860G	SA360G	SA360G GREEN,SA360G YELLOW	SA360G GREEN,SA360G YELLOW,SA360G RED
9	SA360G	SA160G	SA160G GREEN	SA160G GREEN,SA160G RED
10	SB148G	SB248G	SB248G GREEN,SB248G YELLOW	SB248G GREEN,SB248G YELLOW,SB248G RED
11	SB248G	SB348G	SB348G GREEN,SB348G YELLOW	SB348G GREEN,SB348G YELLOW,SB348G RED
12	SB140G	SB240G	SB240G GREEN,SB240G YELLOW	SB240G GREEN,SB240G YELLOW,SB240G RED
13	SB240G	SB340G	SB340G GREEN,SB340G YELLOW	SB340G GREEN,SB340G YELLOW,SB340G RED
14	SB240G	SB348G		SB348G GREEN,SB348G YELLOW,SB348G RED
15	SB868G	SB768G	SB768G GREEN,SB768G YELLOW	SB768G GREEN,SB768G YELLOW,SB768G RED
16	SB768G	SB668G	SB668G GREEN,SB668G YELLOW	SB668G GREEN,SB668G YELLOW,SB668G RED
17	SB860G	SB760G	SB760G GREEN,SB760G YELLOW	SB760G GREEN,SB760G YELLOW,SB760G RED
18	SB760G	SB660G	SB660G GREEN,SB660G YELLOW	SB660G GREEN,SB660G YELLOW,SB660G RED
19	SB760G	SB668G		SB668G GREEN,SB668G YELLOW,SB668G RED
20	SC148G	SC448G	SC448G GREEN,SC448G YELLOW	SC448G GREEN,SC448G YELLOW,SC448G RED
21	SC448G	SC648G	SC648G GREEN,SC648G YELLOW	SC648G GREEN,SC648G YELLOW,SC648G RED
22	SC448G	SC640G		SC640G GREEN,SC640G YELLOW,SC640G RED
23	SC140G	SC440G	SC440G GREEN,SC440G YELLOW	SC440G GREEN,SC440G YELLOW,SC440G RED
24	SC440G	SC640G	SC640G GREEN,SC640G YELLOW	SC640G GREEN,SC640G YELLOW,SC640G RED
25	SC440G	SC648G		SC648G GREEN,SC648G YELLOW,SC648G RED
26	SC868G	SC668G	SC668G GREEN,SC668G YELLOW	SC668G GREEN,SC668G YELLOW,SC668G RED
27	SC668G	SC468G	SC468G GREEN,SC468G YELLOW	SC468G GREEN,SC468G YELLOW,SC468G RED
28	SC668G	SC460G		SC460G GREEN,SC460G YELLOW,SC460G RED
29	SC860G	SC660G	SC660G GREEN,SC660G YELLOW	SC660G GREEN,SC660G YELLOW,SC660G RED
30	SC660G	SC460G	SC460G GREEN,SC460G YELLOW	SC460G GREEN,SC460G YELLOW,SC460G RED
31	SC660G	SC468G		SC468G GREEN,SC468G YELLOW,SC468G RED
32	SD148G	SD348G	SD348G GREEN,SD348G YELLOW	SD348G GREEN,SD348G YELLOW,SD348G RED
33	SD348G	SD448G	SD448G GREEN,SD448G YELLOW	SD448G GREEN,SD448G YELLOW,SD448G RED
34	SD348G	SD440G		SD440G GREEN,SD440G YELLOW,SD440G RED
35	SD140G	SD340G	SD340G GREEN,SD340G YELLOW	SD340G GREEN,SD340G YELLOW,SD340G RED
36	SD340G	SD440G	SD440G GREEN,SD440G YELLOW	SD440G GREEN,SD440G YELLOW,SD440G RED
37	SD768G	SD668G	SD668G GREEN,SD668G YELLOW	SD668G GREEN,SD668G YELLOW,SD668G RED
38	SD668G	SD568G	SD568G GREEN,SD568G YELLOW	SD568G GREEN,SD568G YELLOW,SD568G RED
39	SD668G	SD560G		SD560G GREEN,SD560G YELLOW,SD560G RED
40	SD760G	SD660G	SD660G GREEN,SD660G YELLOW	SD660G GREEN,SD660G YELLOW,SD660G RED
41	SD660G	SD560G	SD560G GREEN,SD560G YELLOW	SD560G GREEN,SD560G YELLOW,SD560G RED
42	SE148G	SE248G	SE248G GREEN,SE248G YELLOW	SE248G GREEN,SE248G YELLOW,SE248G RED
43	SE248G	SE748G	SE748G GREEN,SE748G YELLOW	SE748G GREEN,SE748G YELLOW,SE748G RED
44	SE248G	SE740G		SE740G GREEN,SE740G YELLOW,SE740G RED

45	SE140G	SE240G	SE240G GREEN,SE240G YELLOW	SE240G GREEN,SE240G YELLOW,SE240G RED
46	SE240G	SE740G	SE740G GREEN,SE740G YELLOW	SE740G GREEN,SE740G YELLOW,SE740G RED
47	SE240G	SE748G		SE748G GREEN,SE748G YELLOW,SE748G RED
48	SE868G	SE768G	SE768G GREEN,SE768G YELLOW	SE768G GREEN,SE768G YELLOW,SE768G RED
49	SE768G	SE268G	SE268G GREEN,SE268G YELLOW	SE268G GREEN,SE268G YELLOW,SE268G RED
50	SE768G	SE260G		SE260G GREEN,SE260G YELLOW,SE260G RED
51	SE860G	SE760G	SE760G GREEN,SE760G YELLOW	SE760G GREEN,SE760G YELLOW,SE760G RED
52	SE760G	SE260G	SE260G GREEN,SE260G YELLOW	SE260G GREEN,SE260G YELLOW,SE260G RED
53	SE760G	SE268G		SE268G GREEN,SE268G YELLOW,SE268G RED
54	SF148G	SF248G	SF248G GREEN,SF248G YELLOW	SF248G GREEN,SF248G YELLOW,SF248G RED
55	SF248G	SF748G	SF748G GREEN,SF748G YELLOW	SF748G GREEN,SF748G YELLOW,SF748G RED
56	SF248G	SF340G		SF340G GREEN,SF340G YELLOW,SF340G RED
57	SF140G	SF240G	SF240G GREEN,SF240G YELLOW	SF240G GREEN,SF240G YELLOW,SF240G RED
58	SF240G	SF340G	SF340G GREEN,SF340G YELLOW	SF340G GREEN,SF340G YELLOW,SF340G RED
59	SF868G	SF768G	SF768G GREEN,SF768G YELLOW	SF768G GREEN,SF768G YELLOW,SF768G RED
60	SF768G	SF268G	SF268G GREEN,SF268G YELLOW	SF268G GREEN,SF268G YELLOW,SF268G RED
61	SF768G	SF660G		SF660G GREEN,SF660G YELLOW,SF660G RED
62	SF860G	SF760G	SF760G GREEN,SF760G YELLOW	SF760G GREEN,SF760G YELLOW,SF760G RED
63	SF760G	SF660G	SF660G GREEN,SF660G YELLOW	SF660G GREEN,SF660G YELLOW,SF660G RED
64	SG148G	SG248G	SG248G GREEN,SG248G YELLOW	SG248G GREEN,SG248G YELLOW,SG248G RED
65	SG248G	SG348G	SG348G GREEN,SG348G YELLOW	SG348G GREEN,SG348G YELLOW,SG348G RED
66	SG140G	SG240G	SG240G GREEN,SG240G YELLOW	SG240G GREEN,SG240G YELLOW,SG240G RED
67	SG240G	SG740G	SG740G GREEN,SG740G YELLOW	SG740G GREEN,SG740G YELLOW,SG740G RED
68	SG240G	SG348G		SG348G GREEN,SG348G YELLOW,SG348G RED
69	SG868G	SG768G	SG768G GREEN,SG768G YELLOW	SG768G GREEN,SG768G YELLOW,SG768G RED
70	SG768G	SG668G	SG668G GREEN,SG668G YELLOW	SG668G GREEN,SG668G YELLOW,SG668G RED
71	SG860G	SG760G	SG760G GREEN,SG760G YELLOW	SG760G GREEN,SG760G YELLOW,SG760G RED
72	SG760G	SG260G	SG260G GREEN,SG260G YELLOW	SG260G GREEN,SG260G YELLOW,SG260G RED
73	SG760G	SG668G		SG668G GREEN,SG668G YELLOW,SG668G RED
74	SH148G	SH248G	SH248G YELLOW,SH248G GREEN	SH248G YELLOW,SH248G RED
75	SH248G	SH340G		SH340G GREEN,SH340G RED
76	SH140G	SH240G	SH240G GREEN,SH240G YELLOW	SH240G GREEN,SH240G YELLOW,SH240G RED
77	SH240G	SH340G	SH340G GREEN	SH340G GREEN,SH340G RED
78	SH360G	SH260G	SH260G GREEN,SH260G YELLOW	SH260G GREEN,SH260G YELLOW,SH260G RED
79	SH260G	SH160G	SH160G GREEN	SH160G GREEN,SH160G RED
80	SH260G	SH168G		SH168G GREEN,SH168G RED
81	SH160G	SG860G	SG860G GREEN,SG860G YELLOW	
82	SH168G	SG868G	SG868G GREEN,SG868G YELLOW	
83	SB268G	SB168G	SB168G GREEN	SB168G GREEN,SB168G RED
84	SB268G	SB160G		SB160G GREEN,SB160G RED
85	SB168G	SA868G	SA868G GREEN,SA868G YELLOW	
86	SB160G	SA860G	SA860G GREEN,SA860G YELLOW	
87	SB348G	SB648G	SB648G GREEN,SB648G YELLOW	SB648G GREEN,SB648G YELLOW,SB648G RED
88	SB348G	SB646G		SB646G YELLOW,SB646G RED
89	SB648G	SB748G	SB748G GREEN,SB748G YELLOW	SB748G GREEN,SB748G YELLOW,SB748G RED
90	SB646G	SB748G		SB748G GREEN,SB748G YELLOW,SB748G RED
91	SB260G	SB160G	SB160G GREEN	SB160G GREEN,SB160G RED
92	SB340G	SB640G	SB640G GREEN,SB640G YELLOW	SB640G GREEN,SB640G YELLOW,SB640G RED
93	SB340G	SB642G		SB642G YELLOW,SB642G RED
94	SB640G	SB740G	SB740G GREEN,SB740G YELLOW	SB740G GREEN,SB740G YELLOW,SB740G RED
95	SB642G	SB740G		SB740G GREEN,SB740G YELLOW,SB740G RED
96	SB668G	SB368G	SB368G GREEN,SB368G YELLOW	SB368G GREEN,SB368G YELLOW,SB368G RED
97	SB668G	SB366G		SB366G YELLOW,SB366G RED
98	SB368G	SB268G	SB268G GREEN,SB268G YELLOW	SB268G GREEN,SB268G YELLOW,SB268G RED
99	SB366G	SB268G		SB268G GREEN,SB268G YELLOW,SB268G RED
100	SB748G	SB848G	SB848G GREEN	SB848G GREEN,SB848G RED
101	SB748G	SB840G		SB840G GREEN,SB840G RED
102	SB848G	SC148G	SC148G GREEN,SC148G YELLOW	
103	SB840G	SC140G	SC140G GREEN,SC140G YELLOW	
104	SB660G	SB360G	SB360G GREEN,SB360G YELLOW	SB360G GREEN,SB360G YELLOW,SB360G RED
105	SB660G	SB362G		SB362G YELLOW,SB362G RED
106	SB360G	SB260G	SB260G GREEN,SB260G YELLOW	SB260G GREEN,SB260G YELLOW,SB260G RED
107	SB362G	SB260G		SB260G GREEN,SB260G YELLOW,SB260G RED

108	SB740G	SB840G	SB840G GREEN	SB840G GREEN,SB840G RED
109	SC468G	SC168G	SC168G GREEN	SC168G GREEN,SC168G RED
110	SC468G	SC160G		SC160G GREEN,SC160G RED
111	SC168G	SB868G	SB868G GREEN,SB868G YELLOW	
112	SC160G	SB860G	SB860G GREEN,SB860G YELLOW	
113	SC648G	SC848G	SC848G GREEN	SC848G GREEN,SC848G RED
114	SC648G	SC840G		SC840G GREEN,SC840G RED
115	SC848G	SD148G	SD148G GREEN,SD148G YELLOW	
116	SC840G	SD140G	SD140G GREEN,SD140G YELLOW	
117	SC460G	SC160G	SC160G GREEN	SC160G GREEN,SC160G RED
118	SC460G	SC168G		SC168G GREEN,SC168G RED
119	SC640G	SC840G	SC840G GREEN	SC840G GREEN,SC840G RED
120	SC640G	SC848G		SC848G GREEN,SC848G RED
121	SD368G	SD168G	SD168G GREEN	SD168G GREEN,SD168G RED
122	SD168G	SC868G	SC868G GREEN,SC868G YELLOW	
123	SD448G	SD548G	SD548G GREEN,SD548G YELLOW	SD548G GREEN,SD548G YELLOW,SD548G RED
124	SD448G	SD546G		SD546G YELLOW,SD546G RED
125	SD548G	SD648G	SD648G GREEN,SD648G YELLOW	SD648G GREEN,SD648G YELLOW,SD648G RED
126	SD546G	SD648G		SD648G GREEN,SD648G YELLOW,SD648G RED
127	SD360G	SD160G	SD160G GREEN	SD160G GREEN,SD160G RED
128	SD360G	SD168G		SD168G GREEN,SD168G RED
129	SD160G	SC860G	SC860G GREEN,SC860G YELLOW	
130	SD440G	SD540G	SD540G GREEN,SD540G YELLOW	SD540G GREEN,SD540G YELLOW,SD540G RED
131	SD440G	SD542G		SD542G YELLOW,SD542G RED
132	SD540G	SD640G	SD640G GREEN,SD640G YELLOW	SD640G GREEN,SD640G YELLOW,SD640G RED
133	SD542G	SD640G		SD640G GREEN,SD640G YELLOW,SD640G RED
134	SD568G	SD468G	SD468G GREEN,SD468G YELLOW	SD468G GREEN,SD468G YELLOW,SD468G RED
135	SD568G	SD466G		SD466G YELLOW,SD466G RED
136	SD468G	SD368G	SD368G GREEN,SD368G YELLOW	SD368G GREEN,SD368G YELLOW,SD368G RED
137	SD466G	SD368G		SD368G GREEN,SD368G YELLOW,SD368G RED
138	SD648G	SD748G	SD748G GREEN	SD748G GREEN,SD748G RED
139	SD748G	SE148G	SE148G GREEN,SE148G YELLOW	
140	SD560G	SD460G	SD460G GREEN,SD460G YELLOW	SD460G GREEN,SD460G YELLOW,SD460G RED
141	SD560G	SD462G		SD462G YELLOW,SD462G RED
142	SD460G	SD360G	SD360G GREEN,SD360G YELLOW	SD360G GREEN,SD360G YELLOW,SD360G RED
143	SD462G	SD360G		SD360G GREEN,SD360G YELLOW,SD360G RED
144	SD640G	SD740G	SD740G GREEN	SD740G GREEN,SD740G RED
145	SD640G	SD748G		SD748G GREEN,SD748G RED
146	SD740G	SE140G	SE140G GREEN,SE140G YELLOW	
147	SE268G	SE168G	SE168G GREEN	SE168G GREEN,SE168G RED
148	SE268G	SE160G		SE160G GREEN,SE160G RED
149	SE168G	SD768G	SD768G GREEN,SD768G YELLOW	
150	SE160G	SD760G	SD760G GREEN,SD760G YELLOW	
151	SE748G	SE848G	SE848G GREEN	SE848G GREEN,SE848G RED
152	SE748G	SE840G		SE840G GREEN,SE840G RED
153	SE848G	SF148G	SF148G GREEN,SF148G YELLOW	
154	SE840G	SF140G	SF140G GREEN,SF140G YELLOW	
155	SE260G	SE160G	SE160G GREEN	SE160G GREEN,SE160G RED
156	SE260G	SE168G		SE168G GREEN,SE168G RED
157	SE740G	SE840G	SE840G GREEN	SE840G GREEN,SE840G RED
158	SE740G	SE848G		SE848G GREEN,SE848G RED
159	SF268G	SF168G	SF168G GREEN	SF168G GREEN,SF168G RED
160	SF168G	SE868G	SE868G GREEN,SE868G YELLOW	
161	SF748G	SF848G	SF848G GREEN	SF848G GREEN,SF848G RED
162	SF848G	SG148G	SG148G GREEN,SG148G YELLOW	
163	SF260G	SF160G	SF160G GREEN	SF160G GREEN,SF160G RED
164	SF260G	SF168G		SF168G GREEN,SF168G RED
165	SF160G	SE860G	SE860G GREEN,SE860G YELLOW	
166	SF340G	SF640G	SF640G GREEN,SF640G YELLOW	SF640G GREEN,SF640G YELLOW,SF640G RED
167	SF340G	SF642G		SF642G YELLOW,SF642G RED
168	SF640G	SF740G	SF740G GREEN,SF740G YELLOW	SF740G GREEN,SF740G YELLOW,SF740G RED
169	SF642G	SF740G		SF740G GREEN,SF740G YELLOW,SF740G RED
170	SF660G	SF360G	SF360G GREEN,SF360G YELLOW	SF360G GREEN,SF360G YELLOW,SF360G RED

171	SF660G	SF362G		SF362G YELLOW,SF362G RED
172	SF360G	SF260G	SF260G GREEN,SF260G YELLOW	SF260G GREEN,SF260G YELLOW,SF260G RED
173	SF362G	SF260G		SF260G GREEN,SF260G YELLOW,SF260G RED
174	SF740G	SF840G	SF840G GREEN	SF840G GREEN,SF840G RED
175	SF740G	SF848G		SF848G GREEN,SF848G RED
176	SF840G	SG140G	SG140G GREEN,SG140G YELLOW	
177	SG268G	SG168G	SG168G GREEN	SG168G GREEN,SG168G RED
178	SG268G	SG160G		SG160G GREEN,SG160G RED
179	SG168G	SF868G	SF868G GREEN,SF868G YELLOW	
180	SG160G	SF860G	SF860G GREEN,SF860G YELLOW	
181	SG348G	SG648G	SG648G GREEN,SG648G YELLOW	SG648G GREEN,SG648G YELLOW,SG648G RED
182	SG348G	SG646G		SG646G YELLOW,SG646G RED
183	SG648G	SG748G	SG748G GREEN,SG748G YELLOW	SG748G GREEN,SG748G YELLOW,SG748G RED
184	SG646G	SG748G		SG748G GREEN,SG748G YELLOW,SG748G RED
185	SG260G	SG160G	SG160G GREEN	SG160G GREEN,SG160G RED
186	SG668G	SG368G	SG368G GREEN,SG368G YELLOW	SG368G GREEN,SG368G YELLOW,SG368G RED
187	SG668G	SG366G		SG366G YELLOW,SG366G RED
188	SG368G	SG268G	SG268G GREEN,SG268G YELLOW	SG268G GREEN,SG268G YELLOW,SG268G RED
189	SG366G	SG268G		SG268G GREEN,SG268G YELLOW,SG268G RED
190	SG748G	SG848G	SG848G GREEN	SG848G GREEN,SG848G RED
191	SG748G	SG840G		SG840G GREEN,SG840G RED
192	SG848G	SH148G	SH148G GREEN,SH148G YELLOW	
193	SG840G	SH140G	SH140G GREEN,SH140G YELLOW	
194	SG740G	SG840G	SG840G GREEN	SG840G GREEN,SG840G RED

Appendix B

UML Class Diagrams

B.1 Introduction

This appendix presents the UML class diagrams and the relationship diagrams between the classes for the control table generator programs. UML class diagram outlines the classes and provides more insight regarding the structure of programs. Relationship diagram illustrates all interactions between the classes.

B.2 UML Class Diagrams

A class represents the common features, such as, attributes and operations of the object in C# language [54]. The attributes are also known as properties and operations also known as methods [48]. Classes can be referred as the blueprints of the object that is used by the program to create an instance of the object [48].

The Unified Modelling Language (UML) class diagram is a formal presentation of a class in a structured manner. UML diagram outlines the structure of the class rather than describing the behaviour of the class [54]. These class diagram lists the attributes and operations of the class in a rectangular table.

The section B.4 and B.5 of this appendix contains all the UML class diagrams used by the control table generator programs. Each classes defines all its attributes and operations.

B.3 Class Relationships

The relationship class diagrams shows the static structure of a system that includes all the classes of the system and the interactions between them [54]. This program utilised some of the relationships between the classes for transferring data within these classes. These relationships are: Inheritance, Association, Aggregation and composition.

The program mainly uses four types of relationships between the classes. They are:

1. Inheritance
2. Association
3. Aggregation
4. Composition

Section B.6 provides the complete relationship class diagram of conventional approach and compositional approach.

B.3.1 Inheritance

Inheritance is one of the most commonly used class relationship. The common attributes and operations are often grouped together and described in the parent class subsequently the child classes are implemented that are able to inherits from the parent class [49]. The parent class also known as base class and child class is also known as derived class. The use of inheritance makes the program more elegant and easier to code. It enables many extra useful features of OOP for the program.

The inheritance diagram of the program using conventional approach is given in Figure B.1 which shows the three signalling elements of a station layout. These three elements, *Signal*, *Point* and *Track*, that are child classes of the *Element*, the parent class.

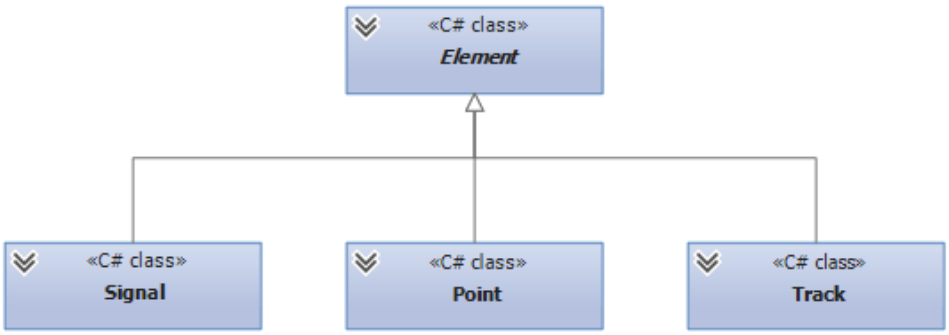


Figure B.1: Inheritance diagram of conventional approach

The inheritance diagram of the program using compositional approach is given in Figure B.2 which shows the additional four sub-sections of a station layout. These four sub-sections, *SingleRoute*, *LoopNetwork*, *SingleCrossover* and *DoubleCrossover*, that are also child classes of the *Element* the parent class.

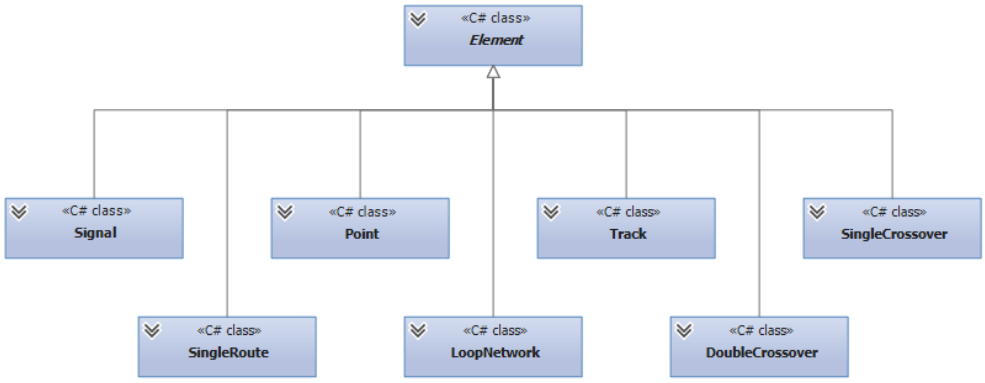


Figure B.2: Inheritance diagram of compositional approach

B.3.2 Association

Generally association is defined as a relationship between objects that allows the objects to communicate with each other [54]. An association is a "using" relationship among objects [55]. In other words, this relationship between classes allows one object to use the data from another object. However, in this relationship each object have

its own lifetime and there is no owner between each other hence the objects can be created and destroyed independently [55].

An association is represented by a single arrow in relationship diagram. The association relationship between two or more objects called object connection or link that denotes a path of communication between them [55]. An association also describes the multiplicity among the objects by indicating one-to-one, one-to-many, or many-to-many relationships between the objects [54]. Multiplicity value is noted in the diagram by specifying the minimum and maximum value where an asterisk mean many (zero or more). An example of association is shown in Figure B.3.

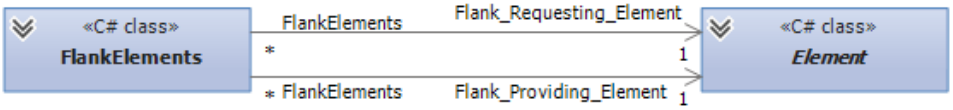


Figure B.3: Association from the Program

B.3.3 Aggregation

Aggregation is a special form of association between two or more objects where an object consists of the entire or a composite of the other object that work together [49, 54]. Aggregation represents a whole-part relationship among the objects [54]. Moreover, in this relationship each object have its own lifetime and there exist an ownership relation between the object hence the whole object can be exist without the part object [55].

The responsibility of aggregate relationship is such that the owner object send message to the appropriate parts [54]. The implementation of the aggregation can be identified in the program where an instance of one object contains instances of some other objects [54].

An aggregation is represented by a line with a hollow diamond at the end of the object that owns the other object in relationship diagram. Aggregation is also capable of one-to-one, one-to-many, or many-to-many relationship between the participating objects [55]. An example of aggregation is shown in Figure B.4.

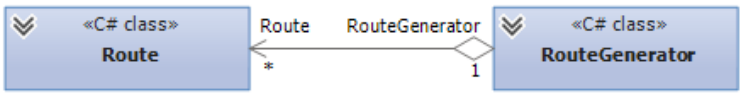


Figure B.4: Aggregation from the Program

B.3.4 Composition

Composition is a stricter form of aggregation that also represents a *whole-part* relationship between the objects [54]. In composition one of the object represents the *whole* system and other object represents the *part* of the system. However, in composition the life cycle of the *part* is depended on the *whole* that is the owner in this relationship [55].

When the *whole* object is created in the program then the *part* objects are created as well in the program. Furthermore, when the *whole* object is destroyed in the program then the *part* objects are destroyed as well in the program [54]. Therefore the composition is a strong type of aggregation which is referred to as a "death" relationship [55]. A composition is represented by a line with a solid diamond at the end of the object that owns the other object in relationship diagram. An example composition is shown in Figure B.5.

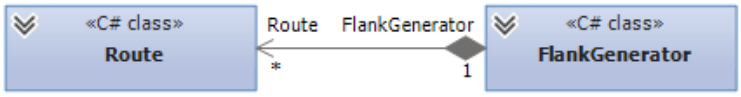


Figure B.5: Composition from the Program

B.4 Class Diagram of Conventional Approach

Element is a parent class. Figure B.6 shows the UML class diagram of the class *Element* using the conventional approach. This class includes the *ElementName* and linking properties where *ElementName* is a *String* type and linking properties are *Element* type hence it is a self-referencing class. Because it is a parent class the

virtual attributes are stated here and the *override* attributes are implemented in child classes.

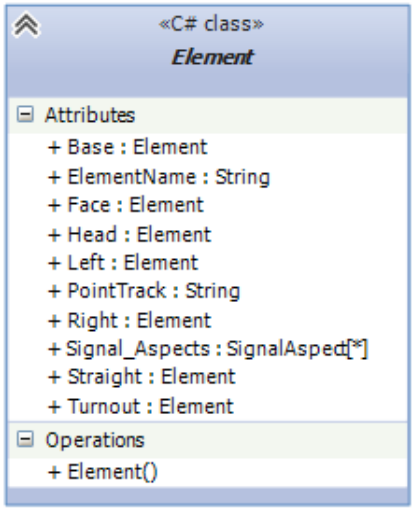


Figure B.6: Element class diagram for conventional approach

SignalAspect is an enumeration that lists all valid signal aspects, they are, GREEN, RED and YELLOW according to three aspects signalling system. Figure B.7 shows the diagram of *SignalAspect* enumeration.

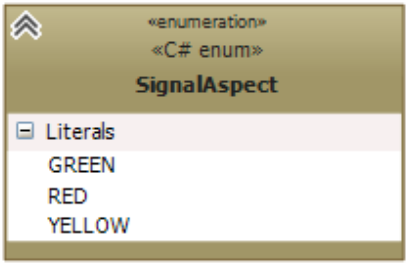


Figure B.7: SignalAspect class diagram

Signal is a child class of the *Element* class. It inherits the *ElementName* from the *Element* class and linking properties are overridden attributes. Figure B.8 shows the structure of *Signal* class where *Base* and *Head* are two linking properties. The aspects of the signal are stored as an array of *SignalAspect*.

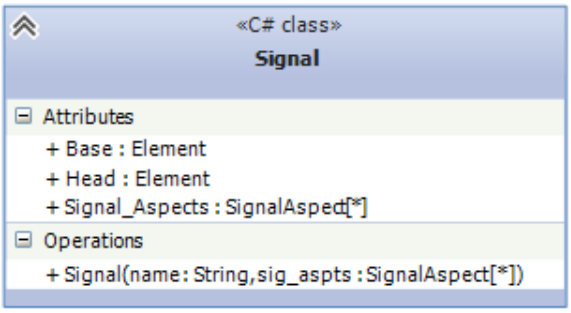


Figure B.8: Signal class diagram

Point is a child class of the *Element* class. It inherits the *ElementName* from the *Element* class and linking properties are overridden attributes. Figure B.9 shows the structure of *Point* class where *Straight*, *Turnout* and *Face* are three linking properties.

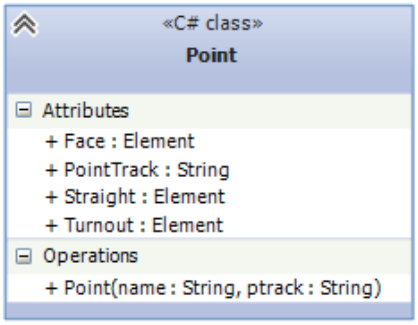


Figure B.9: Point class diagram

Track is a child class of the *Element* class. It inherits the *ElementName* from the *Element* class and linking properties are overridden attributes. Figure B.10 shows the structure of *Track* class where *Left* and *Right* are two linking properties.

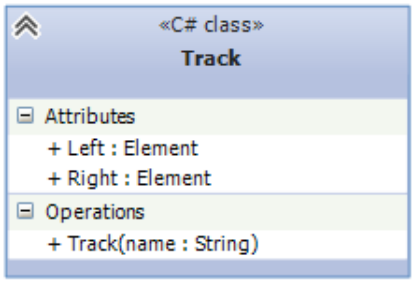


Figure B.10: Track class diagram

Route is a class that contains all the information of the route, for example, the departure signal, destination signal, points in the route, etc. These are the attributes of the *Route* class. Route class also contains the temporary *Elements*, namely *current_ element* and *next_ element* that are used to navigate through the layout. Algorithm evaluates all these information and stores them into the route objects. Figure B.11 gives all the attributes of the class.

Figure B.11: Route class diagram

Figure B.12 provides the structure of the *RouteGenerator* class. This class is responsible for the evaluation of the route information. It uses a number of methods to evaluate the route information that is shown in Figure B.12.

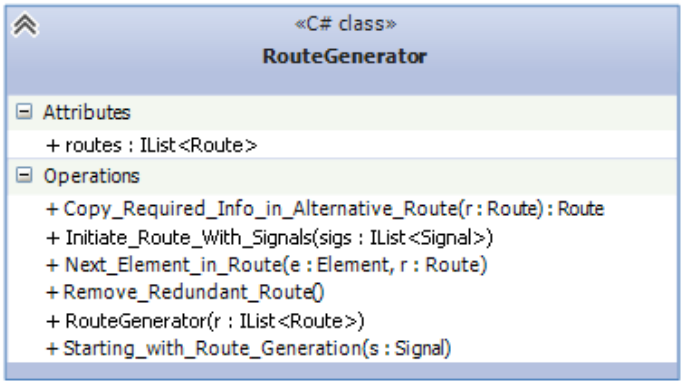


Figure B.12: RouteGenerator class diagram

Figure B.13 provides the structure of the *AspectGenerator* class. This class is responsible for the evaluation of the aspect information. It uses a number of methods to evaluate the aspect information that is shown in Figure B.13.

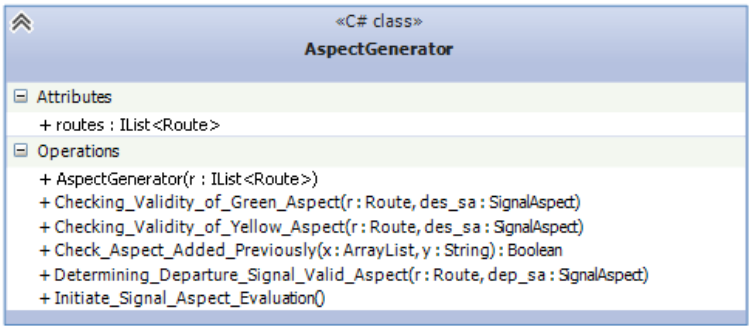


Figure B.13: AspectGenerator class diagram

FlankElements class is used to store the flank requesting and flank providing element. Figure B.14 shows the structure of the *FlankElements* class.

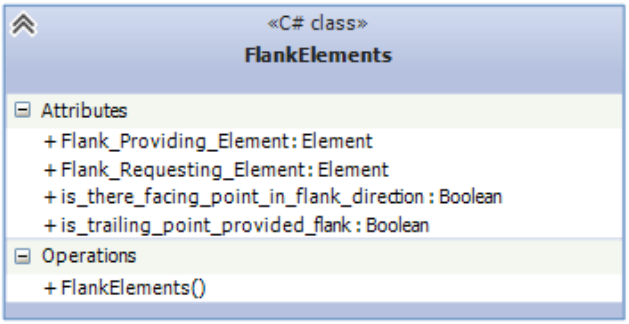


Figure B.14: FlankElement class diagram

Figure B.15 provides the structure of the *FlankGenerator* class. This class is responsible for the evaluation of the flank protection information. It uses a number of methods to evaluate the flank protection information that is shown in Figure B.15.

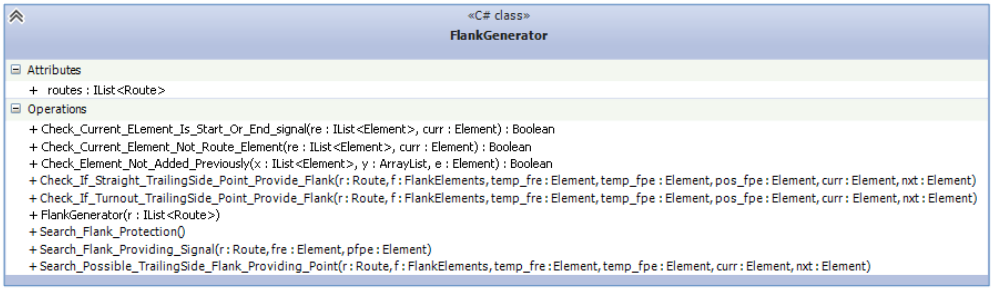


Figure B.15: FlankGenerator class diagram

StationLayout is a class that allows to define all the elements of the layout and enables to link the elements to each other as per the geographical position. Figure B.16 shows the UML class diagram of *StationLayout*.

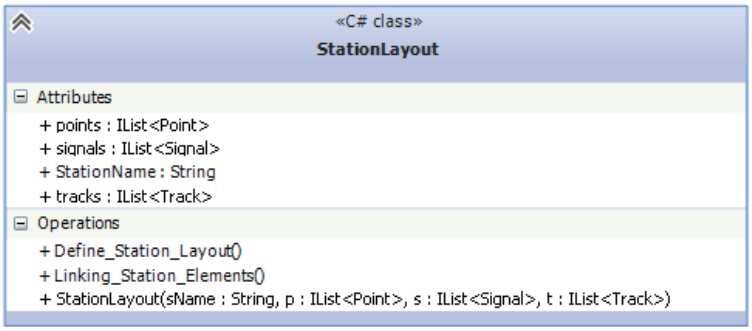


Figure B.16: StationLayout class diagram

B.5 Class Diagram of Compositional Approach

Compositional approach uses four extra software objects to define the sub-sections of the signalling layout, they are: Single Route, Loop Network, Single Crossover and Double Crossover. Therefore four new classes are introduced for control table generator program using compositional approach. Other classes from the conventional approach can be re-used for the compositional approach. Where there were more functionalities required for the evaluation of control table information using compositional approach those classes are shown in this section.

The *Element* class of the compositional approach is much larger than the conventional approach since it defines the virtual attributes of all sub-sections of the program. Figure B.17 provides the UML class diagram of *Element* class using compositional approach.

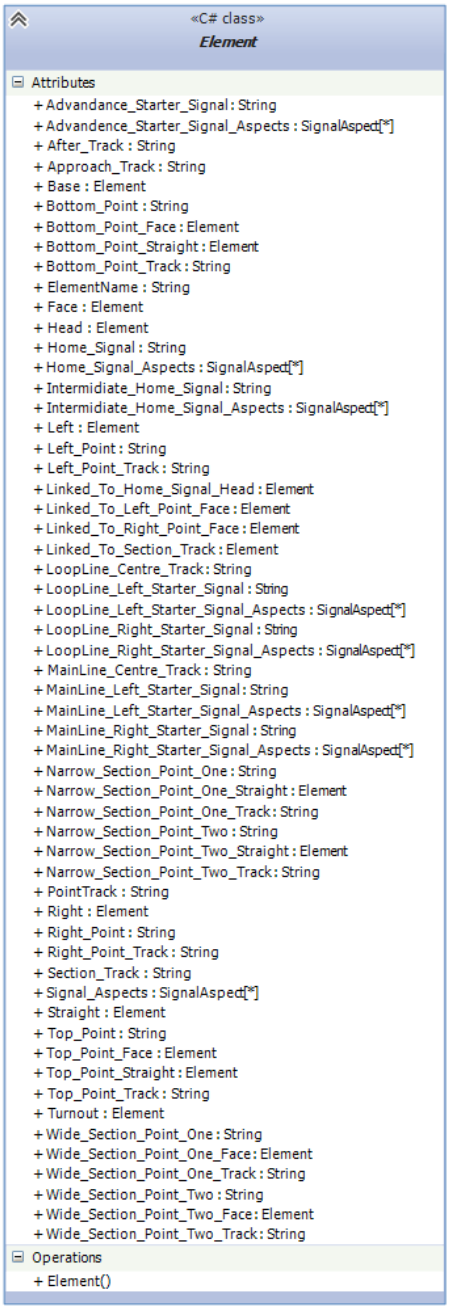


Figure B.17: Element class diagram for compositional approach

The attributes of *SingleRoute* are shown in Figure B.18.

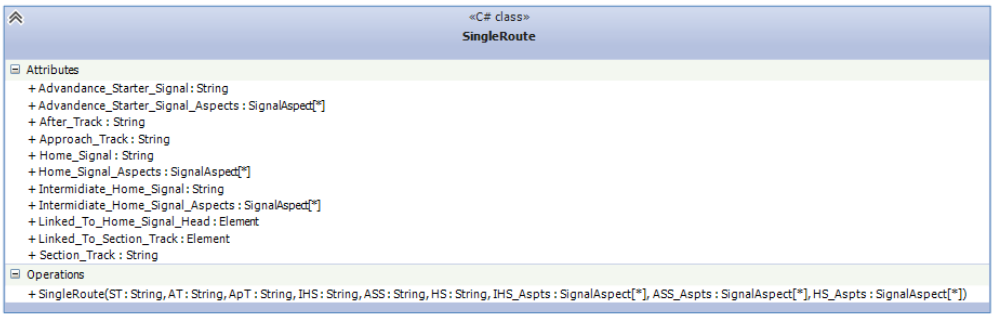


Figure B.18: SingleRoute class diagram

The attributes of *LoopNetwork* are shown in Figure B.19.

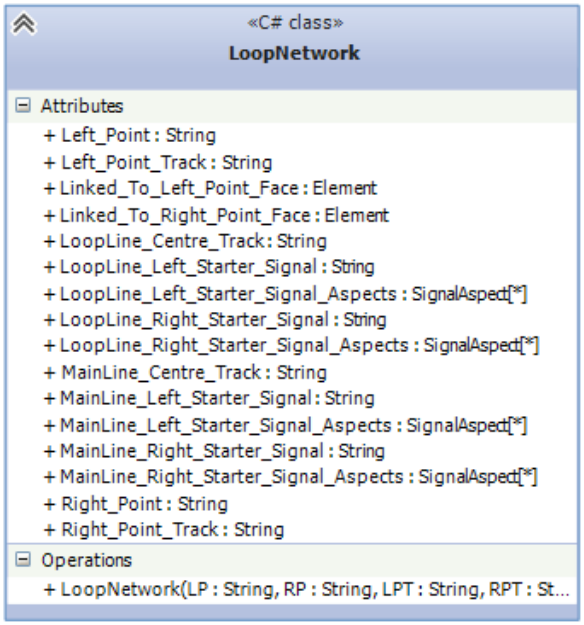


Figure B.19: LoopNetwrok class diagram

The attributes of *SingleCrossover* are shown in Figure B.20.

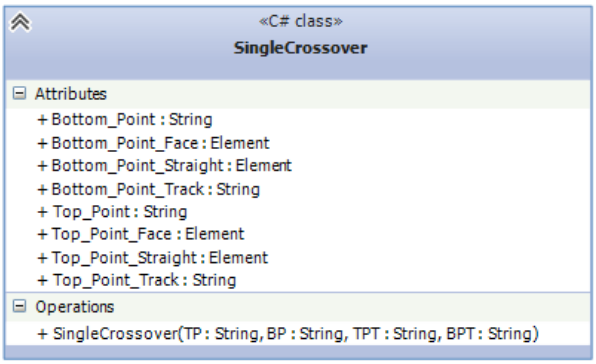


Figure B.20: SingleCrossover class diagram

The attributes of *DoubleCrossover* are shown in Figure B.21.

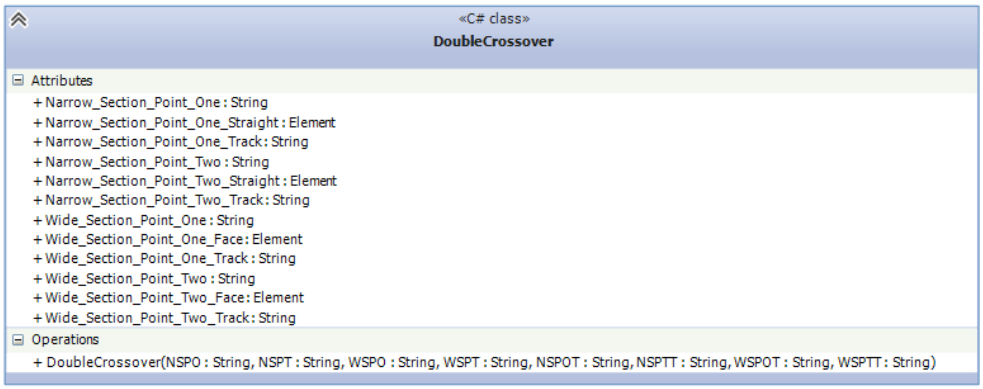


Figure B.21: DoubleCrossover class diagram

RouteGenerator class for compositional approach is given in Figure B.22. The additional operation of the *RouteGenerator* is stated here.

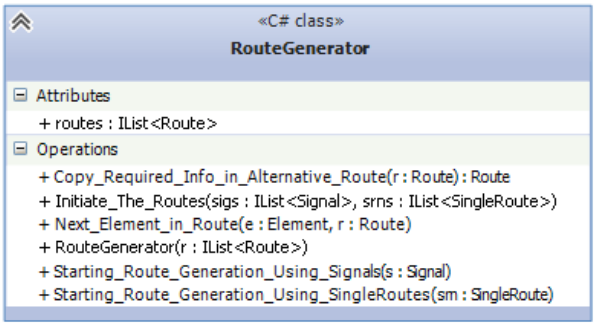


Figure B.22: RouteGenerator class diagram for the compositional approach

FlankGenerator class for compositional approach is given in Figure B.23. The additional operations of the *FlankGenerator* are stated here.

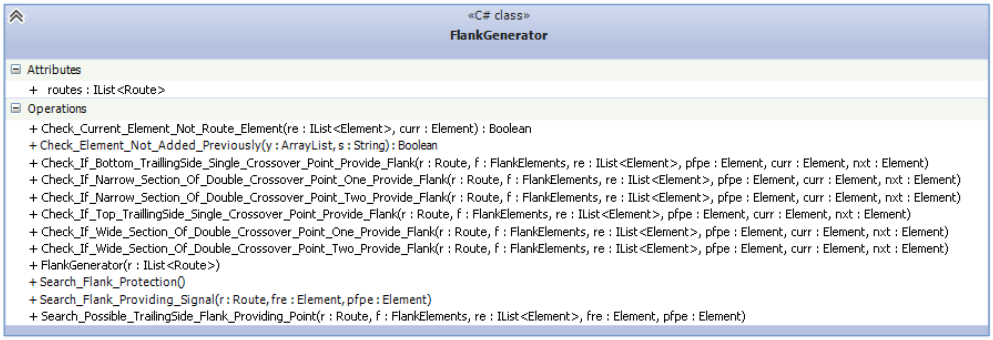


Figure B.23: FlankGenerator class diagram for the compositional approach

B.6 Relationship Diagrams

This section provides the relationship diagrams of classes for conventional approach and compositional approach.

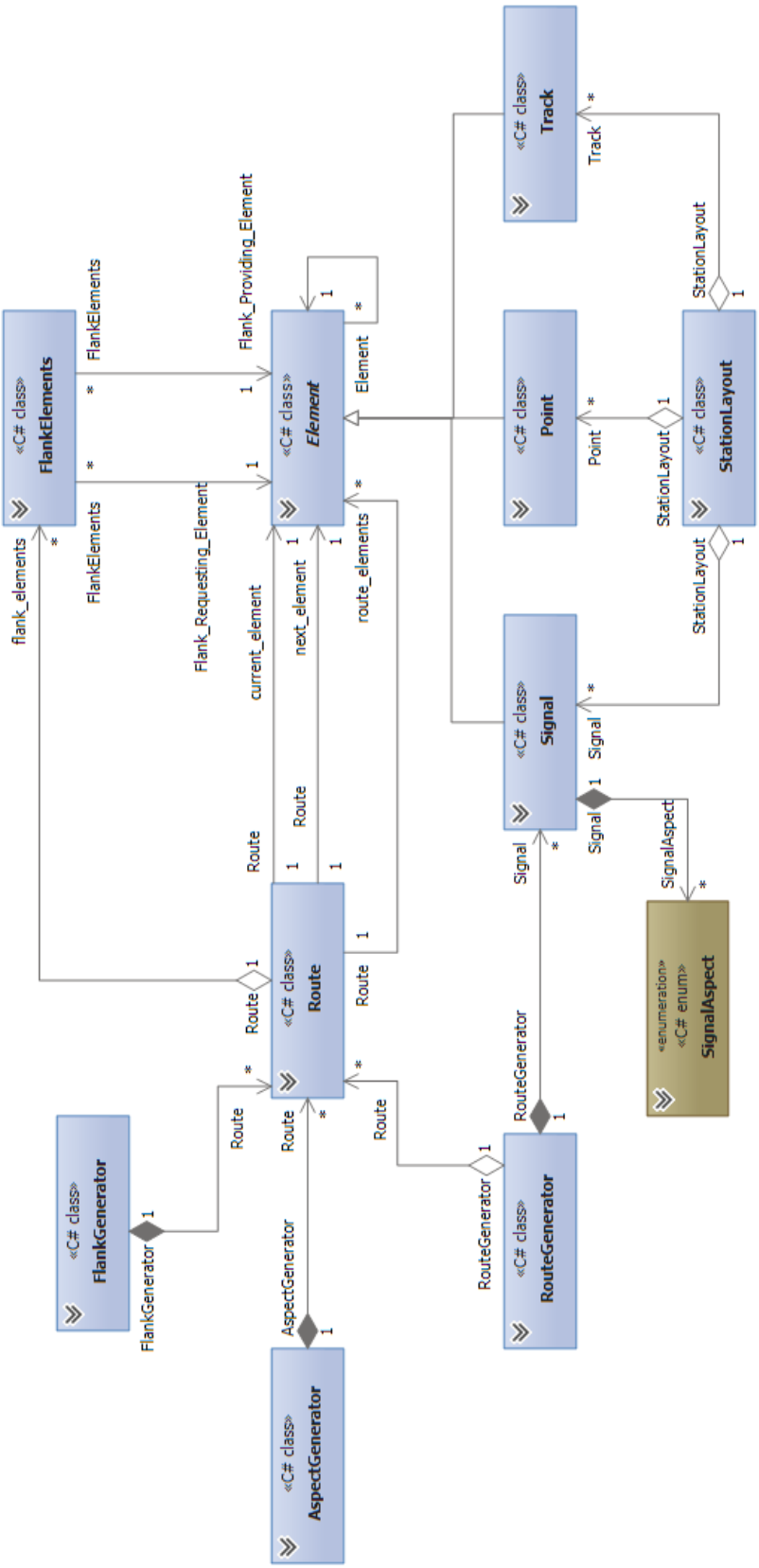


Figure B.24: Relationship class diagram using conventional approach

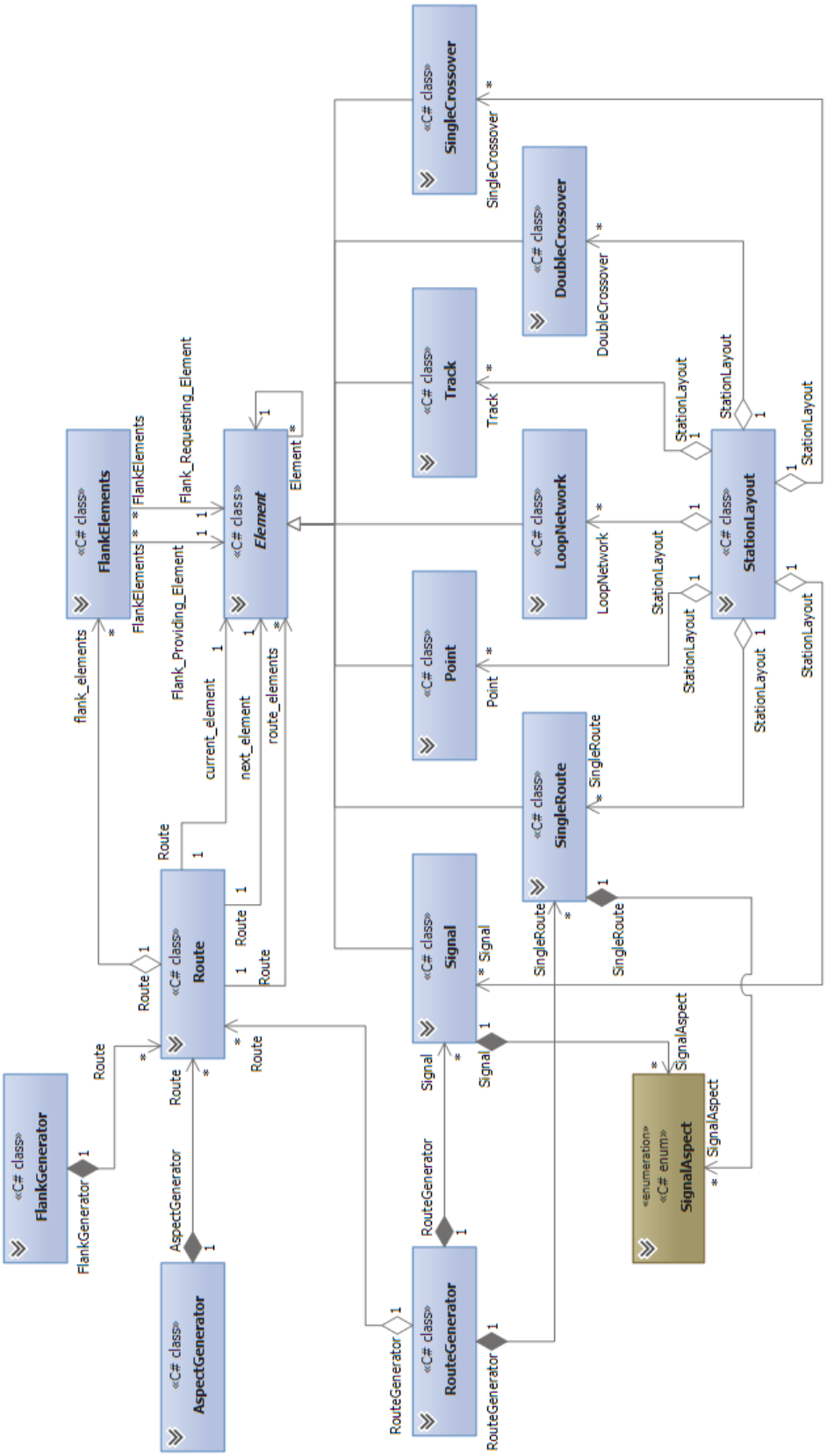


Figure B.25: Relationship class diagram using compositional approach

Appendix C

Program Performance

C.1 Introduction

This appendix provides the performance of the control table generator programs in terms of the various computational resource usage, for instance, CPU execution time and memory usage. The Performance Profiler tool in Visual Studio was used to investigate various performance categories to evaluate the efficiency of the program.

C.2 Execution Time

This section of the appendix contains the execution time of the control table generator programs which were developed using conventional approach and compositional approach. It also illustrates the CPU performance in a graph format for throughout program's execution period.

The performance profiler has some useful features such a the percentage use of the CPU during the execution of the control table generator programs. The percentage of CPU usage is displayed in a graph where the x-axis presents the *Time* in Milliseconds (ms) and the y-axis presents the percentage use of CPU. The Figure C.1 shows the graph of the percentage of CPU usage during the execution time.

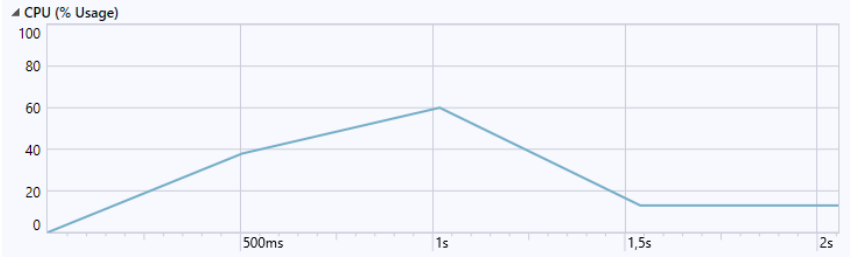


Figure C.1: The percentage of CPU usage during the execution of the program

The timestamp is another important feature that can be extracted from the profiling tool. Two most relevant events from the timestamp for this investigation are the start of the program and the end of the program shown in Milliseconds (ms). The picture in Figure C.2 shows the timestamps of the program.

Mar...	Mark Name	Timestamp
0	Start of Program	0,00
1	AutoMark	0,00
2	AutoMark	506,74
3	AutoMark	1 019,87
4	AutoMark	1 541,44
...	End of Program	2 056,45
5	AutoMark	2 056,45

Figure C.2: The timestamp

The execution time of control table generator programs were determined by utilising the timestamp feature which are recorded and presented in the tabular form. Table C.1 presents the total execution time of the control table generator programs which were developed using different approaches for a number of layouts. Thus the programs execution time of different layouts can be compared between the conventional approach and compositional approach.

Table C.1: Execution time for different layout

Station Layout	Conventional Approach	Compositional Approach
Single Turnout	2056.45ms	2055.78ms
Single Loop	2062.94ms	2036.30ms
Single Crossover	2066.81ms	2039.18ms
Double Crossover	2052.95ms	2059.71ms
Complex	2595.76ms	2598.92ms
Large	4936.98ms	3646.87ms
Line Plan	8099.74ms	3340.16ms

It was understood from a practical point of view the program requires very minimal time to execute. The difference of execution time for conventional approach and compositional approach can be presented in milliseconds.

Hence a single run of the program using the performance profiler is not sufficient to make a judgement on the outcome of the research. Therefore a statistical analysis was conducted using multiple runs of the programs. Appendix D presents the results of the statistical analysis.

C.3 Function Calls

The performance profiler does not provide the most desired feature for this research which is the number of the operations executed by the program that are only related to the algorithm. The profiling tool however is able to provide the number of function calls for the *exe* file.

The Figure C.3a shows the total number of function calls by the control table generator program of conventional approach for the *Single Turnout* station layout while Figure C.3b shows the total number of function calls by the control table generator program of compositional approach for the exact same *Single Turnout* station layout.

Name	Number of Calls
▷ **mscorlib.dll**	**1 358**
◢ **Conv_ctg_single_turnout.exe**	**572**
Conv_ctg_single_turnout.Route.get_route...	56
Conv_ctg_single_turnout.Element.get_Bas...	51
Conv_ctg_single_turnout.Element.get_Rig...	48
Conv_ctg_single_turnout.Element.get_He...	46
Conv_ctg_single_turnout.Element.get_Ele...	40
Conv_ctg_single_turnout.RouteGenerator....	39
Conv_ctg_single_turnout.Element.get_Left()	36
Conv_ctg_single_turnout.FlankGenerator....	28
Conv_ctg_single_turnout.AspectGenerato...	19

(a) The number of function calls in conventional approach

Name	Number of Calls
▷ **mscorlib.dll**	**1 172**
◢ **Comp_ctg_single_turnout.exe**	**279**
Comp_ctg_single_turnout.Ro...	21
Comp_ctg_single_turnout.Asp...	19
Comp_ctg_single_turnout.Asp...	16
Comp_ctg_single_turnout.Ele...	15
Comp_ctg_single_turnout.Ro...	12
Comp_ctg_single_turnout.Asp...	10
Comp_ctg_single_turnout.Ele...	10
Comp_ctg_single_turnout.Fla...	8

(b) The number of function calls in compositional approach

Figure C.3: The number of function calls using both approaches

A table is drafted in order to make a comparison for the number of function calls required by the programs between the conventional and compositional approach for various station layouts. The Table C.2 provides the number of function calls required during the execution of the program for different layouts.

Table C.2: Number of function calls for different layout

Station Layout	Conventional Approach	Compositional Approach
Single Turnout	572	279
Single Loop	883	262
Single Crossover	947	386
Double Crossover	1693	534
Complex	10401	6271
Large	48980	5346
Line Plan	481422	12648

It is apparent in the comparison above that the number of the function calls of conventional approach is much greater than the compositional approach. The value of function calls are directly related to the algorithm of the program hence the higher values of conventional approach is justified.

C.4 Memory Usage

The memory usage is one of the most important features of the determine the efficiency of the program. The performance profiler tool provides a graph with the memory usage of the program throughout the execution of the program. The Figure C.4 depicts the memory usage during the program execution.

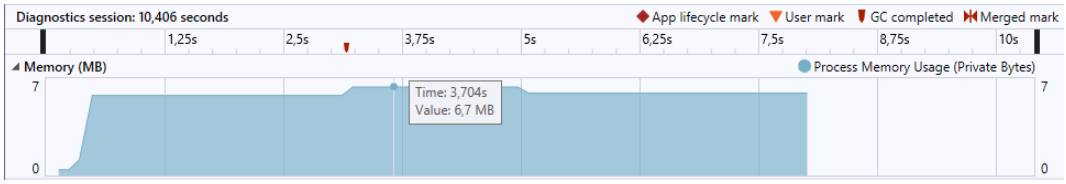


Figure C.4: The Memory Usage

This diagnostic tool provides the amount of memory used in Megabyte (MB) by the program for the entire period that program ran. It is obvious in the Figure C.4 that the maximum usage of the memory during the execution was 6.7MB. The Table C.3 was documented using these graph from the diagnostic tool of the program. This table lists the maximum memory usage of the program for various layouts for conventional approach and compositional approach.

Table C.3: Memory Usage for different layout

Station Layout	Conventional Approach	Compositional Approach
Single Turnout	6.7MB	6.8MB
Single Loop	6.7MB	6.8MB
Single Crossover	6.8MB	6.7MB
Double Crossover	6.8MB	6.8MB
Complex	7.1MB	7.1MB
Large	7.2MB	6.9MB
Line Plan	8.1MB	7.8MB

It is transparent from Table C.3 there are almost no difference in the memory usage in different approaches of the program. The algorithm allocates memories for each object only once in the program and it ensures that no memory duplication of the object occurs during the entire lifetime of the program. The program avoids duplication of memory by extensively making use of *pass-by-reference* software technique for accessing the object between the functions.

Each signalling element is allocated only once by using both approaches, for instance, in conventional approach each signalling element is defined in it's own software object and the composition approach the signalling element is defined within the sub-sectional software object. This justifies that the both approaches use same amount of memory allocation and no compromise was made in use of memory allocation to decrease the CPU execution time.

Appendix D

Conducting Multiple Runs of The Executable Files

D.1 Introduction

This appendix records the execution time of the entire program and the time taken by the algorithm to generate the control tables. A batch file was created to conduct the multiple runs of the executable files of the control table generating program on the various station layouts. Section D.2 to D.8 provide the execution time and the algorithm time of both approaches, conventional and compositional, for all the station layouts used in this research.

D.2 Recording The Times of The Single Turnout Station

The execution time and the algorithm time of the control table generator program using conventional approach and compositional approach for single turnout station are given below.

No.	Conventional Method		Compositional Method	
	Execution Time (ms)	Algorithm Time (ms)	Execution Time (ms)	Algorithm Time (ms)
1	140	17,9926	90	18,989
2	90	16,9908	90	16,992
3	70	14,991	80	14,9918
4	70	14,9916	60	14,9918
5	80	16,9913	70	15,9927
6	80	14,992	70	15,9916
7	70	14,9904	70	15,9918
8	70	15,9919	70	14,9921
9	70	14,9915	80	14,992

10	90	18,9877	80	21,9885
11	80	17,9891	70	14,991
12	70	15,9916	80	15,9913
13	80	23,9859	80	25,9843
14	90	20,9886	80	15,9921
15	70	14,9926	90	16,9912
16	80	15,9922	80	16,9921
17	70	16,9913	80	15,9915
18	80	15,9889	70	15,9928
19	70	15,9906	80	15,9917
20	70	14,9937	80	14,9921
21	80	18,9882	80	16,9912
22	70	14,991	80	15,9911
23	70	14,9911	70	15,9921
24	70	15,9912	70	15,9913
25	70	14,992	70	15,9916
26	80	16,9914	80	14,9818
27	90	14,9924	70	15,9921
28	70	15,9911	80	16,9919
29	70	14,9921	80	15,99
30	70	14,9912	70	15,9936
31	70	15,9918	70	14,9919
32	70	15,9918	80	15,9907
33	80	14,9922	80	15,9914
34	80	15,99	80	14,9923
35	70	14,995	80	22,9871
36	80	14,9914	80	15,9928
37	80	14,9905	70	15,9926
38	70	15,9811	70	15,9915
39	80	15,9916	80	15,99
40	80	15,9917	70	15,9914
41	70	14,991	70	16,9923
42	70	15,9902	80	14,9913
43	80	16,9911	80	15,993
44	70	15,9919	70	15,9917
45	70	14,9919	80	15,9878
46	80	21,9872	80	15,9917
47	80	15,9915	80	15,9906
48	70	15,9924	80	15,992
49	60	15,9922	80	16,9906
50	70	15,9901	80	14,9924
51	70	14,9925	80	16,9901
52	80	21,9885	70	14,9926
53	80	15,9894	70	15,9906
54	80	18,9892	70	15,9918
55	80	20,9888	70	15,9891
56	80	15,9923	80	14,9928
57	70	15,9924	80	16,9896
58	70	16,9895	70	16,9917
59	70	14,9903	70	15,9913
60	70	14,9924	80	15,9916
61	80	18,9884	70	15,989
62	80	15,9902	80	19,9897
63	70	15,9915	80	16,9851
64	70	15,9929	80	16,9885
65	70	16,9906	60	14,9917
66	70	14,9923	70	15,9907
67	70	15,9911	90	15,9931
68	90	21,9873	70	14,9897
69	80	15,9927	80	17,9889
70	80	14,9923	80	15,9904
71	80	19,9897	70	15,9917
72	70	14,9922	80	15,9923
73	70	14,991	80	15,9926
74	80	18,9882	70	14,9922
75	70	15,9911	70	15,9926
76	80	17,9907	80	16,989
77	80	15,9921	90	15,989

78	70	15,9923	70	15,9894
79	70	15,9911	70	14,9924
80	70	15,9919	80	14,9908
81	80	15,9916	80	15,9911
82	80	16,9913	70	15,992
83	80	15,9903	80	14,9919
84	80	15,9919	90	15,9919
85	80	15,9913	70	15,9917
86	70	14,993	70	15,9919
87	70	15,9923	70	15,9906
88	80	15,9924	70	15,9857
89	70	14,992	70	15,9919
90	80	20,9902	80	15,9881
91	80	15,9913	80	14,9915
92	80	14,9916	70	14,9921
93	80	15,99	70	15,9895
94	80	15,9907	70	15,9924
95	70	14,9917	70	15,992
96	80	17,9901	80	19,987
97	80	14,9922	80	18,9887
98	90	17,9901	70	15,9925
99	60	15,9926	80	14,993
100	70	15,9903	70	14,991

D.3 Recording The Times of The Single Loop Station

The execution time and the algorithm time of the control table generator program using conventional approach and compositional approach for single loop station are given below.

No.	Conventional Method		Compositional Method	
	Execution Time (ms)	Algorithm Time (ms)	Execution Time (ms)	Algorithm Time (ms)
1	90	15,627	170	15,6224
2	100	15,6241	110	31,2495
3	90	15,6164	80	15,6281
4	80	15,6245	90	15,6253
5	80	15,6269	80	15,6243
6	80	15,6249	80	15,6243
7	70	15,6259	80	15,627
8	80	15,6251	80	15,6236
9	70	15,6277	80	15,622
10	100	31,252	80	15,6258
11	80	15,6252	80	15,6247
12	70	15,6253	80	15,6246
13	80	15,6279	140	15,6231
14	100	15,6236	60	15,6245
15	70	31,2496	70	15,6258
16	60	15,6247	80	15,6244
17	60	15,6245	100	15,6242
18	70	31,2486	90	15,6264
19	60	15,6238	110	15,6236
20	80	15,6252	150	15,6244
21	60	31,2497	100	15,6251
22	80	15,6246	80	15,6266
23	80	15,6259	80	15,6254
24	70	15,626	80	15,6262
25	60	15,6281	90	15,6248
26	80	31,2512	80	15,625
27	80	15,6248	80	15,627
28	70	15,6239	70	15,6267

29	80	31,2487	80	15,6275
30	80	15,6248	80	15,6244
31	70	15,6284	90	15,6249
32	80	15,6249	80	15,6248
33	100	15,6277	60	15,6241
34	70	15,6285	80	15,6238
35	100	15,6247	70	15,624
36	90	15,6244	80	15,6275
37	60	15,6252	90	15,6256
38	70	15,6236	80	15,6251
39	100	31,2489	80	15,6254
40	90	15,6249	60	15,6263
41	70	15,6245	70	15,6252
42	80	15,6256	80	15,6253
43	60	15,6243	80	15,6246
44	70	15,6175	90	15,6242
45	80	15,6284	100	15,6267
46	100	31,2491	80	15,6234
47	80	15,6259	80	15,6239
48	80	15,6273	70	15,6237
49	80	15,6239	80	15,6277
50	60	15,6245	80	15,6246
51	80	15,624	80	15,6232
52	80	15,6252	80	15,6247
53	80	15,6242	80	15,6152
54	60	15,6248	80	15,6256
55	80	15,6249	70	15,6228
56	80	15,6177	70	15,6182
57	80	15,6252	80	15,6261
58	80	15,6244	80	15,6241
59	70	15,6255	80	15,6228
60	130	46,873	90	15,6236
61	80	15,625	80	15,6244
62	70	15,6251	80	15,6249
63	70	15,624	80	31,2485
64	100	15,6235	80	15,6251
65	80	31,2474	90	15,6293
66	80	15,6223	80	15,624
67	70	15,6236	80	15,624
68	80	15,6276	100	31,2481
69	80	15,6248	80	15,6256
70	80	15,6252	110	15,6235
71	80	15,6221	100	15,6237
72	60	15,6276	80	15,6241
73	80	15,6246	80	15,6265
74	80	15,625	80	15,6268
75	80	15,6243	80	15,6236
76	90	31,2504	100	15,624
77	80	15,6168	90	15,6262
78	80	15,6257	70	15,6262
79	80	15,6246	90	15,6245
80	80	15,6126	80	15,6259
81	80	15,6244	90	15,6255
82	60	15,625	90	15,6167
83	70	31,2514	90	15,6258
84	60	31,2497	110	31,2512
85	80	15,6252	80	15,6248
86	80	15,6271	70	15,6238
87	80	15,6262	80	15,6237
88	80	15,6272	80	31,25
89	80	15,6242	80	15,6239
90	80	15,6251	80	15,6257
91	80	15,6245	80	31,2485
92	80	15,6246	90	15,624
93	80	15,625	80	15,6245
94	80	31,248	90	15,6261
95	80	15,6242	90	15,6251
96	80	15,6265	80	15,6241

97	80	15,6264	140	31,2499
98	70	15,6254	80	15,6273
99	80	15,6246	100	15,6252
100	80	15,6266	100	15,626

D.4 Recording The Times of The Single Crossover Station

The execution time and the algorithm time of the control table generator program using conventional approach and compositional approach for single crossover station are given below.

No.	Conventional Method		Compositional Method	
	Execution Time (ms)	Algorithm Time (ms)	Execution Time (ms)	Algorithm Time (ms)
1	130	20,9847	90	15,6189
2	110	26,9949	90	15,6242
3	70	17,9912	80	31,2496
4	80	17,9904	70	15,6256
5	70	17,9904	70	15,6278
6	80	17,9893	130	46,8736
7	80	16,9911	70	15,6253
8	80	16,9875	70	15,6256
9	80	17,9898	110	15,6252
10	70	16,9906	90	15,6241
11	80	17,9906	80	15,625
12	170	16,9911	70	15,6183
13	100	22,9869	80	15,6244
14	90	16,9911	100	15,6236
15	70	16,9906	70	15,6253
16	70	18,9893	80	15,6238
17	80	16,99	80	15,6251
18	150	22,9875	80	15,6264
19	80	17,9892	80	15,6248
20	70	17,9902	60	15,6251
21	80	17,9906	80	15,6239
22	80	16,9905	80	15,6257
23	80	16,991	80	15,6242
24	80	16,991	70	15,6258
25	100	20,9883	60	15,6257
26	90	19,9889	80	15,628
27	90	34,5906	80	15,6245
28	70	16,9915	70	15,626
29	80	16,9916	60	15,625
30	80	16,9917	70	15,6236
31	80	18,9907	60	15,6267
32	80	18,9894	80	15,6241
33	80	17,9903	60	15,6251
34	70	16,9913	60	15,6233
35	80	16,9919	60	15,6244
36	70	17,9909	80	15,6247
37	80	17,9905	60	15,6241
38	80	16,9894	70	15,6249
39	90	16,9891	60	15,6262
40	80	17,9911	60	15,6249
41	80	16,9917	80	15,6251
42	70	16,9898	60	15,6388
43	70	16,9911	80	15,6239
44	80	23,987	80	15,6244
45	80	19,9902	70	15,6046

46	80	17,989	80	15,6236
47	80	17,9917	80	15,6241
48	70	16,9911	60	15,6247
49	90	26,9824	80	15,6241
50	70	16,9912	70	15,6273
51	80	17,9885	80	15,6241
52	90	18,9901	60	15,6273
53	70	17,9901	60	15,6267
54	70	17,99	80	15,6248
55	80	16,9916	60	15,6283
56	80	17,9902	60	15,6241
57	70	16,9903	80	15,6241
58	80	17,9899	80	15,6257
59	90	17,9908	100	15,6276
60	70	16,9917	70	15,6241
61	70	17,9897	60	15,6257
62	80	16,9907	60	15,6262
63	80	16,9906	60	15,6272
64	30	20,989	60	15,6249
65	80	17,9909	70	15,6252
66	70	17,9887	80	15,6249
67	70	16,9912	60	15,6245
68	80	16,9896	60	15,6249
69	80	16,9909	60	15,6286
70	80	32,9815	70	15,6256
71	90	25,9838	60	15,6299
72	70	19,989	80	15,6244
73	70	16,9913	80	15,6264
74	70	17,9884	70	15,6253
75	80	17,9899	90	46,8735
76	80	15,9916	80	15,626
77	60	15,6262	60	15,6282
78	60	15,6273	80	15,6286
79	80	15,6247	80	15,6206
80	70	16,9912	70	15,6245
81	70	16,9902	80	15,6251
82	70	16,9883	80	15,6249
83	80	18,9881	80	15,6256
84	80	16,9912	80	15,6253
85	80	17,9905	80	15,6265
86	60	15,622	80	15,6246
87	30	17,9893	80	15,6253
88	80	15,6265	80	15,6273
89	80	15,6244	70	15,6255
90	60	15,6242	80	15,6275
91	60	15,6251	60	15,6249
92	80	16,9913	90	46,8735
93	70	16,9893	60	15,6259
94	70	15,991	60	15,6245
95	80	18,9902	60	15,6238
96	70	18,9892	60	15,6247
97	80	17,9875	60	15,6247
98	60	15,6275	80	15,6244
99	80	15,6244	100	15,6248
100	80	15,625	80	15,6264

D.5 Recording The Times of The Double Crossover Station

The execution time and the algorithm time of the control table generator program using conventional approach and compositional approach for double crossover station are given below.

No	Conventional Method		Compositional Method	
	Executive Time (ms)	Algorithm Time (ms)	Executive Time (ms)	Algorithm Time (ms)
1	160	23,9863	280	19,99
2	90	26,0813	90	18,9884
3	90	21,988	90	19,9883
4	90	25,9844	80	19,9883
5	90	20,9867	100	23,9863
6	100	21,9877	80	17,9898
7	70	18,9903	230	18,9889
8	70	19,9866	120	18,9901
9	70	20,9869	90	17,9902
10	220	27,983	110	21,9874
11	110	26,9842	90	17,9905
12	100	22,9877	70	16,9908
13	90	28,9832	80	16,9918
14	300	29,982	90	20,9885
15	90	22,9857	90	18,9897
16	90	23,9861	70	15,9914
17	100	26,9823	100	23,9865
18	80	21,9853	120	19,9891
19	120	23,9865	80	19,9886
20	90	21,9872	100	18,9889
21	90	24,9853	90	21,9871
22	90	22,9864	110	25,9839
23	100	28,983	90	16,9904
24	110	19,989	100	19,9876
25	80	22,9875	190	16,9909
26	140	20,9876	80	19,9869
27	100	25,9853	100	18,9898
28	100	24,9855	70	16,9906
29	80	22,9882	80	16,9903
30	100	21,9883	70	16,9909
31	90	26,9852	210	17,9883
32	90	18,9885	90	18,9896
33	80	19,9888	80	17,9929
34	30	19,985	90	16,9886
35	90	19,9889	70	15,9896
36	170	23,986	90	15,9921
37	90	24,9833	80	22,9871
38	100	26,9836	70	15,9895
39	110	29,9821	70	15,9912
40	90	21,9856	80	15,9897
41	90	19,9876	70	15,9914
42	90	22,9872	80	15,9915
43	100	28,981	90	18,9896
44	130	30,9834	80	16,9897
45	100	27,9832	80	16,9912
46	120	20,9886	80	15,9919
47	90	20,9883	80	16,9897
48	100	30,9811	70	16,9906
49	140	25,9835	100	26,9826
50	140	21,9851	90	18,9891
51	90	23,985	80	18,99
52	100	22,986	90	25,9851
53	90	27,9828	80	15,992
54	80	19,9892	90	16,9901

55	120	24,9835	100	22,9867
56	120	28,9819	90	18,9882
57	110	25,9848	90	19,9878
58	90	21,9874	80	17,9871
59	100	24,9853	100	20,9878
60	100	30,9826	120	19,9878
61	140	55,909	100	26,9818
62	110	24,9852	220	15,9921
63	100	23,9866	120	20,9878
64	100	40,3521	230	152,4824
65	120	19,9889	180	18,9854
66	90	22,9863	110	21,9869
67	90	19,9895	30	24,9857
68	80	19,9895	80	20,9885
69	80	19,9888	80	18,9895
70	230	22,9868	240	20,9887
71	90	23,9855	90	15,9923
72	90	19,9893	80	23,985
73	80	19,9893	90	18,9889
74	100	24,985	90	18,9899
75	100	21,9872	80	18,9896
76	80	18,9891	100	20,9882
77	80	18,9897	130	65,0293
78	80	18,9871	220	16,9896
79	90	23,9862	80	20,9871
80	110	32,9814	100	20,988
81	100	22,9864	80	16,9911
82	150	75,0178	90	26,9842
83	90	26,9839	100	21,9737
84	100	25,9844	90	15,991
85	110	28,9818	70	16,9912
86	80	18,9891	120	17,99
87	70	18,9901	80	16,9908
88	80	18,9901	80	15,989
89	90	22,9866	100	15,9923
90	70	19,9885	80	16,9889
91	90	19,9889	90	19,9898
92	80	18,9888	80	17,9891
93	90	23,9868	90	18,9894
94	80	19,9892	90	16,9919
95	80	19,9882	90	19,989
96	70	18,99	250	16,9897
97	80	19,989	70	15,9913
98	90	27,9837	90	20,9892
99	70	18,9899	110	22,9864
100	90	20,9872	70	15,9907

D.6 Recording The Times of The Complex Station

The execution time and the algorithm time of the control table generator program using conventional approach and compositional approach for the complex station are given below.

No.	Conventional Method		Compositional Method	
	Execution Time (ms)	Algorithm Time (ms)	Execution Time (ms)	Algorithm Time (ms)
1	200	31,2466	380	33,9806
2	100	31,2381	90	31,2496
3	100	31,2496	90	15,6251
4	80	31,2493	100	15,623
5	110	28,9844	80	31,2469

6	90	31,2493	70	15,626
7	80	31,2483	80	15,6263
8	90	31,2509	100	31,2497
9	100	31,2492	90	31,251
10	100	28,9837	80	31,2467
11	80	25,9854	80	31,25
12	110	62,4969	120	31,2484
13	110	37,9764	110	31,2486
14	110	36,1757	110	31,2526
15	80	31,2506	100	31,2528
16	70	31,2488	70	31,2488
17	90	31,2504	80	15,6252
18	100	28,9838	80	31,2374
19	110	31,2485	90	31,2482
20	80	27,9839	90	31,2505
21	90	26,9848	100	31,2504
22	80	31,2505	90	31,2496
23	80	31,252	100	31,2482
24	100	31,2485	110	31,2489
25	80	31,2498	70	31,2494
26	90	31,2506	90	31,2431
27	110	31,2512	100	31,249
28	90	31,2489	90	31,2491
29	140	31,2443	90	31,2481
30	100	31,2512	100	31,2536
31	100	31,2508	100	31,25
32	110	31,2499	90	31,2508
33	70	31,2493	80	31,251
34	110	30,9818	80	31,2515
35	100	34,7224	90	31,2529
36	90	31,2367	90	31,2485
37	100	40,1996	100	31,2514
38	80	31,2511	90	29,9833
39	90	31,2502	100	29,9803
40	80	31,2515	100	31,2492
41	90	31,2522	80	15,6257
42	90	31,2495	90	31,2518
43	80	15,6247	90	31,2484
44	80	31,2494	140	31,2459
45	80	31,2504	80	15,6252
46	90	31,2498	80	15,623
47	90	31,2495	90	31,2473
48	100	31,2487	90	29,98
49	100	31,2502	80	31,2413
50	90	31,2492	80	31,2507
51	100	34,7224	80	15,6246
52	140	46,8741	100	31,2477
53	100	31,2492	90	31,2493
54	90	15,6291	80	31,249
55	90	31,2503	80	31,2466
56	80	31,25	80	31,2499
57	80	15,6262	70	31,2507
58	80	31,2495	90	15,6277
59	110	31,249	90	29,9831
60	100	31,2488	100	31,2504
61	80	31,2465	80	31,2489
62	80	15,6248	80	31,2476
63	110	31,249	100	31,2477
64	80	31,2496	100	31,2505
65	100	31,2509	90	31,2496
66	170	78,1259	90	31,2479
67	90	31,25	100	31,2502
68	110	31,2486	120	31,2499
69	90	31,2395	100	31,2371
70	90	31,2487	90	15,6249
71	100	46,8738	90	31,2501
72	90	31,2496	80	31,2498
73	80	31,2489	80	31,2518

74	80	31,2492	90	31,2502
75	80	31,2503	70	31,2496
76	110	31,2512	80	15,6239
77	100	31,2521	80	31,2517
78	90	31,251	90	15,6244
79	100	31,2498	80	31,2512
80	70	31,2497	100	31,2488
81	70	46,8763	90	15,6244
82	90	31,2524	100	15,6245
83	100	31,2484	80	31,2469
84	80	31,25	80	15,6236
85	100	46,8768	80	31,2481
86	90	31,2495	100	31,2488
87	100	31,2487	90	15,6251
88	90	31,2492	70	15,6249
89	110	31,2515	90	29,9833
90	80	31,2499	80	31,2329
91	70	31,2502	70	31,253
92	100	46,8768	90	31,247
93	90	31,2497	100	31,2502
94	160	93,7468	90	15,6248
95	80	31,2497	100	15,6263
96	90	31,2509	90	31,2521
97	100	31,2489	90	31,2494
98	100	31,2505	90	31,2383
99	100	31,2484	160	31,2481
100	100	31,2511	90	31,2342

D.7 Recording The Times of The Large Station

The execution time and the algorithm time of the control table generator program using conventional approach and compositional approach for the large station are given below.

No.	Conventional Method		Compositional Method	
	Execution Time (ms)	Algorithm Time (ms)	Execution Time (ms)	Algorithm Time (ms)
1	280	31,2484	130	47,9721
2	130	46,8778	130	53,9678
3	140	49,9694	100	38,9778
4	130	42,9738	110	39,977
5	120	41,9756	100	38,9773
6	110	31,2492	110	38,9774
7	120	62,4985	100	39,977
8	110	46,8735	100	39,977
9	110	46,8735	100	39,9771
10	130	46,8725	90	38,9775
11	130	46,8752	100	38,9774
12	120	46,8754	110	38,9777
13	130	46,8742	100	39,9775
14	140	46,8779	110	39,9768
15	130	46,8741	110	39,9769
16	100	46,8758	100	40,9764
17	130	74,5855	100	39,9772
18	100	37,9778	100	39,9771
19	150	60,9627	100	39,9761
20	120	31,2491	110	40,9752
21	120	53,9673	100	40,9755
22	110	46,874	100	38,9781
23	100	40,9762	180	38,9777
24	140	62,4987	110	40,9754

25	130	40,9767	110	39,9775
26	110	46,8655	170	40,974
27	110	46,873	110	39,9774
28	130	46,8749	100	40,974
29	110	43,9743	100	38,9772
30	130	62,4985	110	40,978
31	110	39,9768	100	41,9744
32	130	62,5012	100	39,9774
33	110	43,9743	100	38,978
34	110	31,2491	110	40,9766
35	110	31,2486	100	47,9725
36	110	46,8779	100	40,9764
37	130	46,8764	100	41,9754
38	130	46,8762	100	38,9748
39	120	62,4976	100	37,9774
40	110	46,8733	110	39,977
41	110	31,2531	110	43,9749
42	130	62,4969	110	38,9774
43	130	46,873	100	37,9774
44	110	46,8722	100	39,9755
45	110	31,2477	110	39,9751
46	140	62,4979	100	39,9771
47	170	78,1178	100	38,9773
48	110	46,8743	120	43,9743
49	120	58,9641	110	39,9759
50	130	46,8745	100	38,9779
51	110	46,8725	100	38,9776
52	210	62,4992	110	41,9742
53	130	46,8777	120	40,9754
54	140	62,4977	100	38,9778
55	120	46,8687	240	63,275
56	170	106,3324	100	38,9776
57	110	46,874	130	50,9698
58	130	46,8738	100	38,9769
59	120	46,8734	100	38,9775
60	130	46,8752	100	38,9778
61	110	46,8772	100	42,9747
62	110	46,8751	110	40,975
63	130	46,8725	110	38,9778
64	100	46,8729	100	39,9774
65	110	31,2458	100	39,9773
66	120	46,8735	110	45,9733
67	200	124,9982	90	38,9775
68	130	46,8752	100	38,9774
69	90	46,8734	100	38,9773
70	120	62,5008	100	39,9756
71	120	46,8716	100	38,9801
72	130	46,8747	120	42,9754
73	130	74,5856	100	39,9769
74	110	31,2503	110	41,9753
75	130	46,8748	100	38,9773
76	130	46,8754	100	38,9776
77	110	31,2497	120	38,9778
78	130	46,875	100	39,9775
79	110	46,8747	100	39,9773
80	130	46,8749	110	41,976
81	110	46,8737	110	39,977
82	130	46,875	110	38,9777
83	110	31,2484	100	39,9779
84	110	31,2491	110	39,9768
85	120	46,8737	100	39,9795
86	140	69,4382	100	39,9769
87	140	31,2502	100	39,9787
88	130	46,8748	100	39,9769
89	120	46,8759	90	38,9776
90	120	46,8735	110	39,9765
91	130	65,9574	90	38,9776
92	130	65,9575	110	39,9761

93	130	46,8758	130	39,9769
94	110	62,4849	100	39,9777
95	110	41,9746	100	38,9754
96	100	31,2544	100	39,9774
97	140	62,4989	110	42,9511
98	130	46,8737	100	39,9768
99	110	31,2493	110	39,9752
100	110	46,8779	120	39,9746

D.8 Recording The Times of The Line Plan

The execution time and the algorithm time of the control table generator program using conventional approach and compositional approach for the line plan are given below.

No.	Conventional Method		Compositional Method	
	Execution Time (ms)	Algorithm Time (ms)	Execution Time (ms)	Algorithm Time (ms)
1	360	96,9312	230	60,9649
2	160	103,9359	130	59,9638
3	160	101,9393	130	53,9683
4	190	106,9379	120	58,9655
5	180	100,9338	120	54,9678
6	280	127,9221	120	55,9663
7	190	116,9286	120	54,9679
8	190	115,9876	110	54,9679
9	280	127,9221	110	52,974
10	170	97,9401	130	55,965
11	170	104,9339	120	52,9691
12	180	99,9375	120	58,9653
13	170	95,9418	130	59,9643
14	170	101,939	120	54,9694
15	160	100,9381	110	54,9869
16	190	125,9234	120	54,9666
17	180	104,9368	120	59,9635
18	160	99,9394	140	58,9645
19	180	108,9345	120	54,9677
20	190	117,9296	120	53,9708
21	180	111,9359	110	53,9691
22	160	98,9393	140	55,9661
23	170	103,9369	130	57,9678
24	180	115,9313	130	54,9688
25	160	98,9394	120	53,9699
26	240	109,9319	120	54,9665
27	170	105,9349	120	53,9675
28	170	111,9296	120	52,9692
29	170	112,9324	120	52,9692
30	170	109,9335	130	64,9617
31	190	107,9343	130	53,9692
32	190	111,933	120	53,9686
33	180	103,9374	120	54,9682
34	190	107,9343	120	54,9677
35	180	111,9335	130	59,9637
36	170	100,9394	120	54,9671
37	180	111,9329	130	58,965
38	190	114,93	120	57,9655
39	170	101,9378	130	58,9652
40	170	104,937	110	52,9693
41	170	104,9371	120	53,9686
42	160	101,9378	120	52,9697
43	170	100,9395	140	55,9658

44	180	110,9343	120	57,9657
45	170	100,9376	120	53,9679
46	170	106,9356	110	53,97
47	180	110,9327	140	57,965
48	170	103,9325	120	52,9685
49	170	116,9303	110	55,9663
50	180	115,9296	120	52,9688
51	170	107,9347	120	58,9654
52	170	108,9346	120	53,9684
53	180	108,9352	120	53,9669
54	180	102,9371	120	54,9721
55	170	116,9303	140	69,9586
56	180	114,9307	120	55,9645
57	170	102,9389	120	52,969
58	200	106,9337	120	54,9656
59	200	123,9251	120	54,9674
60	180	108,9337	130	59,9622
61	170	103,9367	160	52,9693
62	170	110,9364	120	52,9677
63	170	104,937	120	52,9687
64	180	114,9307	130	62,9625
65	170	107,9349	130	58,9657
66	180	120,9261	110	54,9658
67	180	112,9317	130	52,9697
68	180	101,9354	130	53,9683
69	160	104,9363	120	56,9692
70	180	115,9288	120	63,9606
71	180	113,9307	120	53,9684
72	190	106,9334	130	65,9613
73	160	105,9347	120	53,9684
74	170	106,9355	110	52,9678
75	170	109,933	120	53,9718
76	170	112,9317	130	53,9676
77	170	112,9317	130	57,9655
78	170	99,9386	120	52,9688
79	160	98,9415	110	55,9711
80	160	103,9383	120	53,9677
81	170	105,9383	140	58,9639
82	170	100,9376	120	54,9677
83	170	106,9358	200	59,1463
84	170	108,9343	120	54,9537
85	160	102,939	130	54,9702
86	180	117,9261	120	51,9694
87	170	103,9376	110	55,9672
88	170	99,9399	120	53,9681
89	170	100,9371	130	57,965
90	170	104,9384	130	58,9684
91	170	101,9386	110	55,9672
92	170	108,935	120	54,9706
93	170	101,9383	140	63,9624
94	160	100,9389	120	57,9658
95	160	97,9413	120	53,9668
96	170	100,9394	120	56,969
97	170	105,9352	120	56,9663
98	160	105,9354	120	52,9691
99	220	103,9374	130	53,9711
100	160	98,9404	120	55,9697